Boris Shishkov (Ed.)

LNBIP 173

Business Modeling and Software Design

Third International Symposium, BMSD 2013
Noordwijkerhout, The Netherlands, July 8–10, 2013
Revised Selected Papers

Springer

Lecture Notes
in Business Information Processing 173

Boris Shishkov (Ed.)

Business Modeling and Software Design

Third International Symposium, BMSD 2013
Noordwijkerhout, The Netherlands, July 8-10, 2013
Revised Selected Papers

Springer

Volume Editor

Boris Shishkov
Interdisciplinary Institute for Collaboration and Research
on Enterprise Systems and Technology – IICREST
Sofia, Bulgaria
E-mail: b.b.shishkov@iicrest.org

ISSN 1865-1348 e-ISSN 1865-1356
ISBN 978-3-319-06670-7 e-ISBN 978-3-319-06671-4
DOI 10.1007/978-3-319-06671-4
Springer Cham Heidelberg New York Dordrecht London

Library of Congress Control Number: 2014937286

Typesetting: Camera-ready by author, data conversion by Scientific Publishing Services, Chennai, India

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Preface

We observe more and more information system development failures (development projects going over time/budget and low user satisfaction), and it is widely agreed that this is largely due to the lack of adequate underlying business/enterprise models. Such models are crucial for understanding and/or (re-)engineering an enterprise. This is needed, in turn, when (partially) automating enterprise processes by means of software systems. Therefore, software generation should stem from corresponding enterprise modeling. This points to the emerging discipline of enterprise engineering that addresses challenges such as agility and adaptability of enterprises. Further, it is essential that enterprise modeling is appropriately bridged to software design, by aligning enterprise modeling concepts and corresponding software specification concepts. Only such an enterprise-software alignment could actually guarantee that: (a) the software system would be properly integrated in its enterprise context; (b) an enterprise-software traceability would be possible allowing not only for software updates driven by new requirements but also for possible enterprise re-engineering activities, inspired by a goal to better fit the enterprise to the IT platform(s) used; (c) re-use could be an issue, counting on enterprise modeling constructs and software components. All this has been dominant for BMSD, the international symposium on Business Modeling and Software Design, bringing together researchers and practitioners interested in business/enterprise modeling and its relationship to software design, and demonstrating for a third consecutive year (in the 2013 edition) a high quality of papers and presentations as well as a stimulating discussion environment. The theme of BMSD 2013 was: "Enterprise Engineering and Software Generation" and the scientific areas of interest to the symposium were: (a) business models and requirements; (b) business models and services; (c) business models and software; (d) information systems architectures. Further, there were three application-oriented special sessions, namely, a special session on e-Health Services and Technologies, a special session on Future Internet Applications for Traffic Surveillance and Management, and a special session on Intelligent Systems and Business Analysis. These special sessions have brought additional practice-driven value to the symposium.

BMSD 2013 was held in Noordwijkerhout, The Netherlands. The symposium was organized and sponsored by the Interdisciplinary Institute for Collaboration and Research on Enterprise Systems and Technology (IICREST), in cooperation with the Dutch Research School for Information and Knowledge Systems (SIKS), the Center for Telematics and Information Technology (CTIT), Aristotle University of Thessaloniki (AUTH), and AMAKOTA Ltd.

This book contains revised and extended versions of a set of selected BMSD 2013 papers. These papers addressed the above-mentioned challenges, by considering a large number of research topics: from more conceptual ones, such as modeling landscapes, process modeling, declarative business rules, and normalized systems to more practical ones, such as business cases development and performance indicators, from more business-oriented ones, such as value modeling and service systems, to topics related to information architectures.

BMSD 2013 received 56 paper submissions from which 33 papers were selected for publication in the symposium proceedings. From these, 13 papers were selected for a 30-minute oral presentation (full papers), leading to a full paper acceptance ratio of around 23%; this not only shows stable development (the full paper acceptance ratio was 20% in 2012 and 17% in 2011) but also indicates the intention of preserving a high-quality forum for the next editions of the symposium. The eight papers published in the current book were selected from the BMSD 2013 full papers. In all BMSD 2013 selections, a double-blind paper evaluation method was used: Each paper was reviewed by at least two internationally known experts from the BMSD Program Committee.

The high quality of the BMSD 2013 program was enhanced by three keynote lectures, delivered by distinguished guests who are renowned experts in their fields, including (alphabetically): Marco Aiello (University of Groningen, The Netherlands), Kecheng Liu (University of Reading, UK), and Leszek Maciaszek (Wroclaw University of Economics, Poland). Their lectures inspired the participants to gain a deeper understanding of the business/enterprise modeling and software design fields. Further, their active participation in the panel discussion and other discussions provided many participants with valuable feedback on their work.

Inspired by three successful editions of BMSD, namely, Sofia 2011, Geneva 2012, and Noordwijkerhout 2013, we are determined to further develop the symposium, and we hope that you will find these papers interesting and consider them a helpful reference in the future when addressing any of the research areas mentioned here.

February 2014 Boris Shishkov

Symposium Committee

Chair

Boris Shishkov	IICREST, Bulgaria

Program Committee

Hamideh Afsarmanesh	University of Amsterdam, The Netherlands
Marco Aiello	University of Groningen, The Netherlands
Mehmet Aksit	University of Twente, The Netherlands
Antonia Albani	University of St. Gallen, Switzerland
Ognian Andreev	Technical University of Sofia, Bulgaria
Paulo Anita	Delft University of Technology, The Netherlands
Rumen Arnaudov	Technical University of Sofia, Bulgaria
Colin Atkinson	University of Mannheim, Germany
Paris Avgeriou	University of Groningen, The Netherlands
Csaba Boer	TBA, The Netherlands
Boyan Bontchev	Sofia University St. Kliment Ohridski, Bulgaria
Frances Brazier	Delft University of Technology, The Netherlands
Barrett Bryant	University of North Texas, USA
Cinzia Cappiello	Politecnico di Milano, Italy
Jorge Cardoso	University of Coimbra, Portugal
Kuo-Ming Chao	Coventry University, UK
Ruzanna Chitchyan	University of Leicester, UK
Samuel Chong	Capgemini, UK
Dimitar Christozov	American University in Bulgaria - Blagoevgrad, Bulgaria
José Cordeiro	Polytechnic Institute of Setúbal, Portugal
Dumitru Dan Burdescu	University of Craiova, Romania
Joop De Jong	Delft University of Technology, The Netherlands
Jan L.G. Dietz	Delft University of Technology, The Netherlands
Teduh Dirgahayu	Universitas Islam Indonesia, Indonesia
Lyubka Doukovska	Bulgarian Academy of Sciences, Bulgaria
Joaquim Filipe	Polytechnic Institute of Setúbal, Portugal

Chiara Francalanci	Politecnico di Milano, Italy
Boris Fritscher	University of Lausanne, Switzerland
J. Paul Gibson	T&MSP - Telecom & Management SudParis, France
Eduardo Goncalves Da Silva	University of Twente, The Netherlands
Rafael Gonzales	Javeriana University, Colombia
Clever Ricardo Guareis De Farias	University of Sao Paulo, Brazil
Jens Gulden	University of Duisburg-Essen, Germany
Markus Helfert	Dublin City University, Ireland
Philip Huysmans	University of Antwerp, Belgium
Ilian Ilkov	IBM, The Netherlands
Ivan Ivanov	SUNY Empire State College, USA
Dmitry Kan	AlphaSense Inc., Russia
Dimitris Karagiannis	University of Vienna, Austria
Marite Kirikova	Riga Technical University, Latvia
Samuel Kounev	Karlsruhe Institute of Technology, Germany
José Paulo Leal	University of Porto, Portugal
Kecheng Liu	University of Reading, UK
Leszek Maciaszek	Wroclaw University of Economics, Poland
Jelena Marincic	University of Twente, The Netherlands
Michele Missikoff	Institute for Systems Analysis and Computer Science, Italy
Dimitris Mitrakos	Aristotle University of Thessaloniki, Greece
Preslav Nakov	Qatar Computing Research Institute, Qatar Foundation, Qatar
Ricardo Neisse	Fraunhofer IESE, Germany
Bart Nieuwenhuis	University of Twente, The Netherlands
Selmin Nurcan	University Paris 1 Pantheon Sorbonne, France
Olga Ormandjieva	Concordia University, Canada
Sietse Overbeek	University of Duisburg-Essen, Germany
Mike Papazoglou	Tilburg University, The Netherlands
Robert Parhonyi	Inter Access, The Netherlands
Marcin Paprzycki	Polish Academy of Sciences, Poland
Oscar Pastor	Universidad Politécnica de Valencia, Spain
Henderik A. Proper	Public Research Centre – Henri Tudor, Luxembourg
Ricardo Queiros	IPP, Portugal
Jolita Ralyte	University of Geneva, Switzerland
Gil Regev	EPFL/Itecor, Switzerland
Ella Roubtsova	Open University, The Netherlands
Irina Rychkova	University Paris 1 Pantheon Sorbonne, France
Shazia Sadiq	University of Queensland, Australia

Tony Shan	Keane Inc., USA
Valery Sokolov	Yaroslavl State University, Russia
Richard Starmans	Utrecht University, The Netherlands
Cosmin Stoica Spahiu	University of Craiova, Romania
Coen Suurmond	RBK Group, The Netherlands
Bedir Tekinerdogan	Bilkent University, Turkey
Linda Terlouw	ICRIS B.V., The Netherlands
Yasar Tonta	Hacettepe University, Turkey
Roumiana Tsankova	Technical University of Sofia, Bulgaria
Marten van Sinderen	University of Twente, The Netherlands
Maria Virvou	University of Piraeus, Greece
Shin-Jer Yang	Soochow University, Taiwan
Benjamin Yen	University of Hong Kong, China
Fani Zlatarova	Elizabethtown College, USA

Invited Speakers

Marco Aiello	University of Groningen, The Netherlands
Kecheng Liu	University of Reading, UK
Leszek Maciaszek	Wroclaw University of Economics, Poland

Table of Contents

Enterprise Modelling Languages
Just Enough Standardisation?

Marija Bjeković[1,2,3], Hend*erik* A. Proper[1,2,3], and Jean-Sébastien Sottet[1,3]

[1] Public Research Centre Henri Tudor, Luxembourg, Luxembourg
[2] Radboud University Nijmegen, Nijmegen, Netherlands
[3] EE-Team, Luxembourg, Luxembourg*
{marija.bjekovic,erik.proper,jean-sebastien.sottet}@tudor.lu

Abstract. In enterprise modelling, a wide range of models and languages is used to support different purposes. If left uncontrolled, this variety of models and languages can easily result in fragmented perspective on an enterprise, its processes and IT support. A traditional approach to address this problem is to create standard modelling languages that unify and integrate different modelling perspectives, such as e.g. UML, BPMN, and ArchiMate. However, one can observe how, in actual use, the 'standardising' and 'integrating' effect of these languages erodes. This is typically manifested by the emergence of 'variants', 'light weight versions', and extensions of the standard dealing with 'missing aspects'. The empirical data suggests that these 'variants' emerge to compensate the inability of a standard language to aptly fit the needs of specific modelling situations. In this paper, we reconsider the drivers and strategies of modelling language standardisation. Relying on an ongoing research, the paper develops a fundamental understanding of the role of fixed language in the context of conceptual and enterprise modelling. This understanding is then used to analyse the 'variants' in the actual use of a standard process modelling language, and to discuss the potential insights towards its standardisation strategy.

Keywords: model, modelling language, standardisation, modelling pragmatics.

1 Introduction

Enterprise models play an important role in the design and operations of enterprises. They typically represent an enterprise from different perspectives, and are used for various purposes, e.g. to study the current state of an enterprise, analyse problems with regard to the current situation, sketch potential future scenarios, design future states of the enterprise, communicate with stakeholders, manage change, etc. If this plethora of models is left uncontrolled, it may result

* The Enterprise Engineering Team (EE-Team) is a collaboration between Public Research Centre Henri Tudor, Radboud University Nijmegen and HAN University of Applied Sciences (www.ee-team.eu).

B. Shishkov (Ed.): BMSD 2013, LNBIP 173, pp. 1–23, 2014.

in a fragmented view on the enterprise, and thus negatively affect the overall coherence of models. The fact that these partial enterprise models are often expressed in different modelling languages makes the coherence even a greater challenge.

The traditional approach of dealing with this situation is to create a unifying modelling language, such as UML for software design, and ArchiMate for enterprise architecture modelling. The assumption behind is that the fragmentation can be prevented by *a priori* integrating relevant perspectives and concepts within a single *standard* modelling language. However, one can observe how, in actual use, the 'standardising' and 'integrating' effect of these languages erodes. This is typically manifested in terms of local 'dialects' [35,9], 'light weight versions' [35], or several extensions of an existing standard that are intended to deal with 'missing aspects' [15,19,40]. The point is further illustrated by the advent of domain-specific [22] and/or purpose-specific modelling languages [6], which allow for the creation of models that are tuned to the needs of specific domains or purposes.

A more realistic strategy to address fragmentation is to create point-to-point bridges (e.g. [41,55,17]) between the modelling languages used in enterprise modelling. The bridges between the languages are typically established based on the *standard* language definitions, in order to be able to reuse them across different usage contexts. Nonetheless, the phenomena of 'dialectisation' is not limited to unifying languages, but is also reported for many enterprise modelling languages, such as goal-oriented, value-oriented and process modelling languages [2,16,33,43]. This is also manifested in dialect-like variations of the original modelling language (e.g. [58]), use of workarounds (e.g. using ad hoc notes, narratives, and annotations, e.g. [43]) to compensate for the missing elements in the language/tool. It may even go as far as using home-grown, organisation-specific semi-structured languages instead of the standard ones [2,35,31]. This phenomena might question the real potential for reuse of standard language bridges, i.e. the value of bridging languages out of context of their use.

While a non-compliance to modelling standards is typically perceived as undesirable, the reasons underlying the wide spread existence of language 'variants' or 'dialects' are not well understood. The available empirical data suggests that they mainly emerge to compensate for the lack of suitability of the language/tool to aptly fit the needs of specific modelling situations. The arguments underlying the widespread use of e.g. Visio as a modelling tool in practice (e.g. [14,35,43]), as well as the growing research interest in modelling language/tool flexibility (e.g. [32,11]), further strengthen this point.

How similar are these 'variants' of a standard? What are the dimensions of divergence from a standard language? If widespread, should these 'variants'(e.g. light-weight versions used in stakeholder communication [35]) be covered by the original standard? How many of these 'variants' should become part of the standard? Finally, to paraphrase, how much standard language would be enough [58]?

In our view, this calls for reconsidering the drivers of, and approaches applied in, modelling language standardisation. But first and foremost, this requires a deep understanding of the role of modelling language in (conceptual and) enterprise modelling, and of the factors driving its use. We believe that this understanding may provide valuable insights for the scoping and design of modelling languages that are better suited to the practical needs. The present paper aims to contribute to such an understanding from a rather theoretical perspective. Relying on our ongoing research, a fundamental understanding of the role of fixed language in conceptual and enterprise modelling is developed in the paper. This understanding is then used to analyse the 'dialectisation' in the actual use of a standard process modelling language BPMN [39], and to discuss the potential insights towards its standardisation strategy.

The remainder of the paper is organised as follows. Section 2 explores the problem of modelling language standardisation, by confronting the rationales of standardisation to the actual use of standard modelling languages. This section identifies the need to deeply understand the role that modelling language has in conceptual/enterprise modelling, and the factors that determine its added value. Subsequently, the Sections 3 and 4 elaborate our fundamental understanding of this topic. This understanding is then used, in Section 5 to analyse the 'variants' emerging in the actual use of a standard process modelling language, BPMN [39], and to discuss the potential insights towards its standardisation strategy, prior to concluding the paper.

2 What Is the Scope of a Modelling Language Standard?

In the field of enterprise modelling, a wide range of *fixed* modelling languages is being defined and used, while some of them also undergo the process of *standardisation*. Despite the ambition of such standards, available empirical studies, e.g. [2,58,43,35,9,31] pinpoint at their inability to fit the needs of practical modelling situations. In practice, this is typically overcome by the emergence of different 'variants' of the used standard, which alter the original language definition, by *reducing*, *extending* or *adapting* it to a modelling task at hand.

Is such behaviour due to the very nature of complex and generic standards, or to the failure to include all the relevant aspects into a standard language? Or, is it rather tied to the way language users perceive the value of the modelling language (standard or not), and even to their subjective preferences? Could these derivations from a specific standard be prevented in the first place, e.g. by a different design of modelling languages? If not, how should they be dealt with? In this section, we revisit the common rationales of modelling language standardisation in 2.1, and confront them to the actual use of such standard languages in 2.2. We then identify, in 2.3, one of the possible research directions to gain a deeper understanding of this problem, which is explored in the paper.

2.1 Drivers of Modelling Language Standardisation

The potential benefits of a *fixed* definition of a modelling language are well-known. It provides a foundation for the development of tools and automated model manipulations (e.g. analysis, simulation, model transformation, code generation), thus enabling the increase of productivity and diminishing the error-rate of model manipulations.

The effort of *standardisation* of such languages is often driven by the desire to generalise these potential benefits across one or more application areas (e.g. system engineering, software engineering, or process modelling) etc. More precisely, the review of commonly used modelling standards in enterprise modelling [38,39,30] reveals that standardisation is typically driven by the following interrelated rationales:

1. Harmonise and consolidate many similar yet divergent modelling languages for modelling some problem area/system, incorporating the best practices of similar methods/notations [38,39],
2. Standardise model exchange format between the tools implementing the language specification [38,39],
3. Provide the standard way to communicate about the problem/system by different stakeholders, and for different uses of models [38,39],
4. Define a uniform representation for a wide range of uses within some problem area [38,39,30],
5. Integrate different perspectives of a system under study within a single standard language [38,39,30],
6. Provide standardised bridge of the gap between the graphical language and the appropriate execution format [39], etc.

Indeed, the harmonisation and/or consolidation of similar overlapping modelling languages already existing for modelling some problem area (e.g. business processes) is a common goal of many standards. The drivers here are many, e.g. reducing language and tool related learning and training costs, tool market harmonisation, facilitating tool selection for practitioners, providing standard model exchange format for tool interoperability. Obviously, the more generic the standard, the more reuse potential it is likely to have across different application areas.

Additionally, a standard modelling language has the ambition to standardise the communication about some problem area/system between various stakeholders and for many purposes for which modelling is done. For instance, the BPMN specification states the ambition to provide the *common* language and visual notation for both business and technical users. An *a priori* imposed standard vocabulary is meant to avoid frequent meta-discussions on concepts between model stakeholders, and to facilitate knowledge transfer. This is tightly related with the drive of defining a *uniform representation* for a range of *uses* of models within a problem area.

The 'uniformisation' and 'harmonisation' often also entail the *integration* of different perspectives, i.e. models, of the system under study within a single, standard *unifying* language definition. As many overlapping languages exist to model the system from different perspectives, the unifying language consolidates these languages within the single integrated language, to be used instead of these partial languages. UML and ArchiMate are typical cases of such a strategy.

The standardising effect of a unifying language hence also lies in that it *a priori* integrates perspectives for modelling some system. Such a strategy facilitates assuring integration between the models, given that consistency and coherence rules can be embedded in the standard language definition, and tools can *automatically check* these properties.

A standardised and/or integrated language is thus one possible strategy to ensure the return on the modelling effort. We can however observe that drivers underlying standardisation effort are predominantly of technical-economical nature. In our view, the potential benefits of standardisation tend to be overly quickly generalised to the entire range of possibles uses of such a standardised language. Below we discuss different challenges of using standardised languages, based on the available empirical data reporting on an enterprise modelling practice.

2.2 Insights from the Use of Standard Modelling Languages

A key problem in *the use of fixed/standard languages* in enterprise modelling seems to be rooted in the lack of *suitability* of a language for the modelling task at hand.

For instance, the widespread need for simpler and rather informal 'variants' of software and enterprise architecture modelling languages is identified by the practitioners interviewed in [35]. These variants are needed in particular for *stakeholder communication*, which is actually reported as the *primary need by practitioners*, and the need that is the most poorly met by existing architectural languages. Despite the abundance of sophisticated and rather formal architectural languages, practitioners still tend to mostly use UML-based languages. Too much formality, as well as too little support for stakeholder communication are indicated as the main reasons for this. Along the same lines, practitioners raise the need for better tool support for language extensibility, for informal activities such as sketching, for combining models with text, etc.

Similarly, the common use of rather informal 'variants' of general-purpose modelling languages is reported in enterprise modelling practice in e.g. [14,9,31]. The need for relaxed versions/dialects of a generic language such as UML is observed, for example, in enterprise modelling situations whose primary goal is *collective knowledge creation* (e.g. developing vision and strategy, scoping the problem, and high-level business design) [9]. As most stakeholders do not have modelling expertise, the language and tools are required to be simple, intuitive, and corresponding to the natural interaction that occurs in such situations [9].

Besides 'variants' of existing standard modelling languages, the use of 'home-grown' or 'ad-hoc' notations is quite common in enterprise modelling practice [2,35,9]. For instance, the study of the use of conceptual models in enterprise modelling efforts across IBM [2], reports that business analysts and business architects clearly prefer home-grown and semi-structured models over the usual standard languages. This is typical for *exploration phases* "where things are unclear and ambiguous" [2, p. 1304], and where semi-structured models offer flexibility in terms of delayed commitment to syntax, unconstrained development order, evolvable re-factorable metamodel, as well as a *closer fit to the inherent* *way of thinking* in these phases [2]. As argued in [2], these notations emerge through the repeated use in similar modelling situations, and gain more structure over time.

In the cases presented so far, we discussed challenges of using standard languages in situations of model-based communication with rather business stakeholders and business concerns. Moreover, in the case of exploration and collective knowledge creation with stakeholders, their involvement and input is crucial for producing 'good' models. For these communication-oriented purposes, standard languages do not seem to be suitable enough: their inherent complexity (in terms of constructs and embedded syntactic-semantic restrictions) seem to rather represent a burden. We have argued in [6,7] that 'variants' emerging in such situations are in fact purpose-specific variations of the original generic language, which tune the language to the needs of given modelling situations. An extreme case of this tuning are, in our view, these ad-hoc and home-grown notations. Such notations can be seen as the emergent modelling languages, which naturally adapt to the needs of situations in which they are used, and gain structure over the course of recurring use in similar modelling situations.

The need for purpose-specific tuning of the language for a given communication situation is a rather natural principle, and indeed corresponding to the way humans normally use natural language [13]. The standards such as e.g. UML and BPMN have the ambition to define the language and notation which is readily usable for the various purposes in their respective application areas, in particular aiming to provide modelling support for business users and purposes. However, a thorough consideration of how these complex standards can/should accommodate these different purposes is lacking. The practice seems to suggest its response.

The practical use of the BPMN standard [39], widely used in enterprise modelling, is examined in [58,43]. It reveals that, in practice, a very small subset of the BPMN constructs is widely used (cluster of around six concepts), another six concepts being occasionally used, while a huge number of concepts is rather superfluous and extremely rarely applied. It is shown in [58] how these different language subsets are used for different purposes. The basic subset of core BPMN concepts (i.e. task, flow, start and end event) together with lanes and pools is rather observed for the purposes of process documentation, organisational (re)design and process improvement. A slightly richer set of constructs, including gateways and event conditions, is used for e.g. simulation and work flow engineering. Overall, the practical use of BPMN demonstrates much less complexity than the standard specification.

Indeed, the BPMN standard is assessed by [43] as over-engineered and way too complex compared to the practical needs. Should then BPMN *as a standard* include those superfluous and rarely used concepts? What is the rationale and added-value of including them in the scope of a *standard definition*? Should the standard language rather be geared towards the most common use of such language? For instance, organisation design and documentation, knowledge management and continuous process improvement are reported, in [43], as the primary purposes for which BPMN models are used. However, BPMN does not provide sufficient constructs to express all the relevant organisational aspects [53,43]. Constructs such as Pools and Lanes are judged as not sufficient for modelling organisational resources in [53]. This finding corroborates the observation by [43] that process models expressed in BPMN are very often extended with the symbols, and even with other models, that capture organisational resources, organisational structure, data, business rules, risks, resources, documents etc. So, if this need is recurrent in the practical use of BPMN, shouldn't these aspects be covered by the standard language?

An additional challenge is present in the use of standard *unifying languages*. Despite the ambition to *a priori* address the integration problem of different perspectives within a standard, the adaptation and extension of such a language with 'missing aspects' can be observed in its actual use. For instance, ArchiMate [30] was initially designed as the enterprise architecture modelling language, which relates relevant 'architectural domains' within a single language umbrella. In the practical use of ArchiMate, it is possible to observe this drive to *extend* ArchiMate models to include specific concerns (e.g. [19,10]) and/or industry-specific standards (e.g. [4]), to cater for contingencies of a *specific application context*. These extensions and adaptations essentially yield a domain-specific version of the original ArchiMate language.

The challenge here lies in the fact that it is nearly impossible to *a priori* identify all the relevant perspectives that should be part of an integrated/unified language for e.g. enterprise modelling. The relevance of different perspectives is highly context-dependent: different perspectives may be relevant for different industries and enterprises, or even in different transformation projects of the same enterprise. Additionally, over time, new perspectives may become relevant (e.g. cloud, privacy, compliance). At the same time, one can observe how there is a drive to *extend* the ArchiMate *standard* to cover the additional aspects, potentially relevant for enterprise modelling. The move from the ArchiMate 1.0 to the ArchiMate 2.0 standard included two additional aspects, namely motivation and implementation & migration. Further integration between TOGAF and ArchiMate is likely to lead to even more extensions. Moreover, the extensions of a standard with e.g. business policies and rules, are also considered [30]. Should all of these considered extensions become integral part of the ArchiMate language? What are the aspects falling under the competence of the ArchiMate language, and which of them should remain outside the language? Potentially endless extensions of ArchiMate are quite likely to result in a fairly complex and, most probably, over-engineered language, similar to the situation of the BPMN 2.0 standard. In practice, this is likely to result in usability problems, and potentially also in many different simplified or adapted variants of a standard being in place. Whether this is a desirable result of a language standardisation has to be questioned [43].

2.3 Discussion

The discussions so far clearly point, in our opinion, to the challenge of defining the *right* scope of a standard modelling language. How should the practical needs and practical use of a language stand in relation with the standard? What is the added value of including some construct or perspective into a standard language? How should this be decided? Should the scope of a standard language be explicitly limited, and based on which criteria? Going further, more fundamental questions are at stake: What does a standard modelling language seek to standardise: the way of thinking about some class of problems in a problem area, the (visual) representation of models (across their different uses) within a problem area, or model exchange format between the modelling tools? Should the single modelling language 'standardise' all these aspects?

To answer these questions, we argue that it is necessary to clearly understand the role that modelling language plays in modelling, and thus identify the main factors determining its added-value in modelling. This is, in our view, necessary if we want to make scientifically grounded decisions regarding the optimal scope of the fixed/standard modelling language. At the same time, this requires tackling some very fundamental aspects of modelling.

3 What Is Modelling?

The next two sections of the paper present the initial version of our *explanatory* theory, which has the ambition to reach a fundamental understanding of the role of fixed language in enterprise modelling. The focus of the theory is on *modelling pragmatics*, i.e. on the use and value of models and modelling languages in the given context, in dependence on the modelling goals [49]. We thus primarily seek to fundamentally understand the phenomena related to the *use of modelling languages* in different situations and for different purposes of modelling. Such a focus requires us to revisit our understanding of the very act of modelling, and the role that purpose has in it. We provide our understanding of these topics in the present section. Section 4 then discusses the role and potential benefits of a fixed modelling language in an enterprise modelling effort.

3.1 Grounding

We understand models as essentially means of communication about some domain of interest, and process of modelling as communication-driven process led by a pragmatic focus [25]. This view is inspired by different related research tackling the fundamental modelling aspects such as [46,45,18,28,42,49].

3.2 Model Definition

Though different views on models and modelling exist, as well as many different model definitions, here we elaborate reasons for which we propose the following (general) model definition (based on [46,45,18,48]):

> *A* ***model*** *is an artefact acknowledged by an observer as representing some domain for a particular purpose.*

By stating that a model is an **artefact**, we exclude *conceptions* [18] or so-called "mental models" from the scope of this definition. The reason for this restriction is practical: in our field, the primary concern is the model-as-artefact resulting from the modelling act.

Conceptions are *abstractions* of the "world" under consideration, adopted from a certain perspective. They share this property with models. However, a conception resides in the mind of a person holding it, and as such is not directly accessible to another human being. To be able to discuss and agree on its content, the conception has to be externalised. While the conceptions reside in mental space, the models are necessarily *represented* in physical/material space (typically using some system of symbols). This *representation* dimension is crucial for any model, as the value of modelling primarily resides in the utility of the model-as-artefact for some purpose.

This said, we do consider conceptions to be fundamental to modelling. This point is thoroughly discussed throughout this section.

The **observer** in our definition refers to the group of people consisting of model creators and model audience. On one extreme, it can refer to the entire society, on the other extreme, to the individual. Though it may not be the general rule, it is very often the case, in an enterprise modelling context, that model creators are at the same time its audience. The observer is the key element in modelling, as it is only by virtue of the observer's appreciation that some artefact comes to be acknowledged as a model of some domain.

Similarly to [18], we define **domain** as any "part" or "aspect" of the "world" *considered relevant by the observer* in the given modelling context. The "world" here may refer not only to the "real" world, but also to hypothetical or imagined worlds. Even more, the domain of a model can be another model as well.

A model always has a **purpose**. This *purposefulness* dimension is present in most of the model definitions, e.g. [46,45,48]. Although acknowledged as essential dimension of models, the concept of purpose is rarely defined and its role in the entire modelling process is scantly discussed.

In the following, we discuss our view on the role of purpose in modelling, as well as suggest our definition of this concept.

3.3 Centrality of Purpose

A **modelling situation** is *at least* characterised by the wider context in which the modelling takes place (e.g. a particular organisation, project), the involved observer, and the goals of the situation.

The goals of the modelling situation are not necessarily restricted to the goal of producing the model. Particularly, in enterprise modelling, these goals may also refer to organisational learning, achieving consensus on a topic and reaching some commonly agreed knowledge [8,34]. The present discussion will focus on these goals that are relative to the desired model-as-artefact.

The reason why an observer creates a model in the first place is to enable some *usage* of that model (e.g. analysis, sketching, execution, contracting etc.) by its intended *audience* (e.g. business analysts, business decision-makers, enterprise architects, process experts, etc.).

We believe these are crucial dimensions underlying the concept of model purpose. As will be discussed later, these dimensions are heavily interdependent. Their combination determines the desired model qualities [8,34] for the purpose at hand. This is quite important, since the *fitness-for-purpose* determines the *value* of the model for its intended use.

We therefore propose the following definition of the model purpose.

The ***purpose*** *of the model is a combination of the following dimensions:*

(1) the domain which the model should pertain to, and

(2) the intended use of the model by its intended audience.

In line with [45,48], that (although usually implicitly present) the purpose should be made explicit within the modelling process. At least the model creator should be aware of the intended usage and audience of the model.

To explain the central role that purpose has in a modelling act, we will first consider the modelling situation where an observer is an individual, illustrated in Figure 1a. When, in this situation, an observer **O** engages in modelling of some "world" under consideration, s/he judges which aspects of that "world" are relevant for the given modelling situation[1]. This process of selecting the relevant and *abstracting away* from the irrelevant aspects of the "world" yields the observer's *conception of the domain*, $\mathbf{c}_d$.

It is here important to underline that this process of abstracting away from irrelevant aspects is always *relative* to the given modelling situation. It is implicitly or explicitly influenced by the purpose $\mathbf{p}$ of the model-to-be $\mathbf{m}_d$ [46,45,48]. This is depicted as an influence of the purpose $\mathbf{p}$ on the relation *conception of* in the Figure 1. Secondly, but not less importantly, how an observer creates an abstraction is also very much dependent on his/her pre-conceptions [42], brought by its particular social, cultural, educational and professional background. We come back to this point in Section 4.

To externalise the conception $\mathbf{c}_d$ residing in his/her mind, the observer **O** subsequently tries to shape an artefact (i.e. the model-to-be) in such a way that it adequately *represents*, for the purpose $\mathbf{p}$, his/her conception of the domain $\mathbf{c}_d$.

At this point, it should be noted that the observer's understanding of the purpose $\mathbf{p}$ is essentially a conception as well, i.e. the conception of the purpose of the model-to-be $\mathbf{c}_p$. Even more, the observer **O** also forms the conception of the model-to-be, $\mathbf{c}_m$. The modelling process thus actually consists in the observer's

[1] Obviously, the observer's judgement may be influenced by many different factors, e.g. observer's intentions, experience, previous knowledge, interests, etc. Our discussion excludes the from the consideration the potential conscious *political intentions* underlying the observer's judgement.

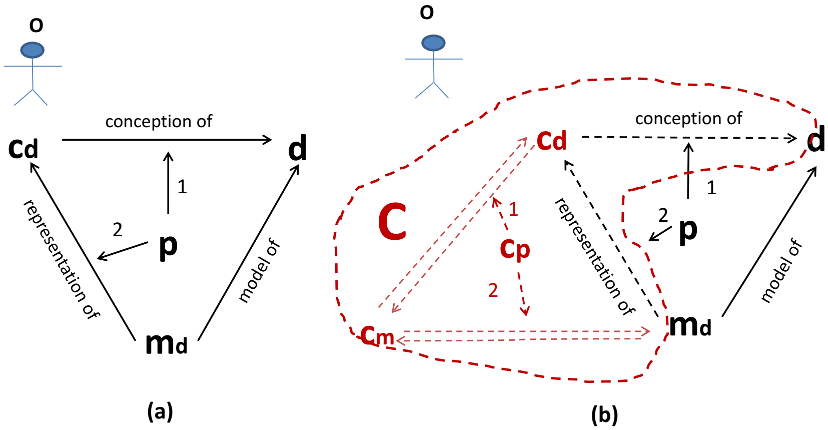


Fig. 1. The act of modelling

gradual alignment of these three conceptions, in parallel with the very shaping of the (model-to-be) artefact. This is illustrated in the Figure 1b.

As this alignment is iterative, the conception of the domain $\mathbf{c}_d$ is not completely stabilised before the artefact (model-to-be) is shaped in a satisfactory manner. It is only at this point that we can speak about the *existence* of the domain **d**, as pointed out in FRISCO [18]. The domain as such does not exist[2] a priori, but emerges in the very act of its modelling, and is being very much shaped by its observer in the given modelling situation.

When the mutual alignment of $\mathbf{c}_d$, $\mathbf{c}_p$ and $\mathbf{c}_m$ is achieved, the artefact comes to be acknowledged as the *representation of* the (conception of the) domain **d** for the purpose **p**. In other words, the observer **O** *acknowledges* that the artefact **m** is a *model of* the domain **d** for the purpose **p**.

Given the modelling situation, the purpose thus determines which features of the domain should be modelled and with which accuracy [45]. This is depicted as the influence of the purpose **p** on the relationship *representation of* in Figure 1. Similar distinction is made in [45], where a model is considered as a special kind of representation, having a *purpose* and a *cost-effectiveness criterion.*

The previous explanation holds for a modelling process where the observer is an individual. In a collaborative modelling situation, a group of **n** human actors is involved in the process of modelling, and is supposed to *jointly* observe some domain **d** and come up with its model $\mathbf{m}_d$, for the purpose **p**. In order to reach a shared view on the domain, the purpose and the model-to-be, the co-alignment of potentially $\mathbf{n}\times\mathbf{3}$ conceptions (i.e. $\mathbf{c}_d$, $\mathbf{c}_m$, $\mathbf{c}_p$ per each of **n** actors) has to take

[2] The term *exist* is used here in the sense of Heidegger's notion of *breaking down*, discussed in [52]. Indeed, "*Heidegger insists that it is meaningless to talk about the existence of objects and their properties in the absence of concernful activity, with its potential for breaking down. What really is is not defined by an objective omniscient observer, nor is it defined by an individual – the writer or computer designer – but rather by a space of potential for human concern and action*" [52, p.37].

place. Indeed, this is considered as a critical step in collaborative modelling, where all the discussions, negotiations and agreement about the model need to take place.

As we have seen, the main factor of the alignment of conceptions is the purpose $\mathbf{p}$, i.e., more precisely, the conception of the purpose by an observer, $\mathbf{c}_p$. As the purpose $\mathbf{p}$ is the main discriminant of the value of created model $\mathbf{m}$, we argue that it should be explicitly considered when creating (and using) models. In other words, the purpose should also be modelled. Explicitly considering and expressing the purpose is important not only to facilitate the (usually implicit) aligning of conceptions, but also to enable the understanding of a model by an observer who was not involved in process of its creation.

Furthermore, and in line with [9,48,29], we argue that the purpose should be the primary driver of shaping (the choice of) modelling language. The modelling language used for modelling should allow expressing the model in such a way that the model is of value for its intended purpose. Some related work embraces this view regarding model and language quality assessment [34], as well as modelling processes [8]. In our research, this position is explored with regards to the definition and use of modelling languages in modelling.

4 Role of Modelling Language in Modelling

Having introduced our fundamental understanding of the modelling act, this section develops our view on the role of modelling language within modelling. We will use such an understanding to discuss the challenges inherent to definition of a fixed modelling language, and more specifically, to its standardisation.

4.1 Grounding the View on Language

In our view, the language consists of a system of symbols whose primary function, i.e. *raison d'être*, is to act as an instrument of human communication and action. Language is thus *used* to formulate and communicate human conceptions of various aspects of the "world", in various communities and communicative circumstances.

This view on language grounds in the body of knowledge of cognitive linguistics [23], functional perspective on language [13,12,52], and semiotics [18,47]. It is adopted in our study of modelling languages, as the phenomena we are interested in go beyond the isolated study of linguistic code, and puts forward its *use* within different modelling situations. We look at the extent to which a fixed/standardised modelling language allows to effectively formulate and communicate conceptions in a given modelling situation, and how it can be designed to better suit this need.

4.2 Elements of Modelling Language Definition

Traditionally, a modelling language is defined in terms of *abstract syntax*, *concrete syntax* and *semantics*.

The **abstract syntax** of a modelling language defines the modelling constructs and rules for their combination when creating models. The abstract syntax of visual modelling languages is usually represented using metamodels.[3]

The **concrete syntax** or *notation* deals with the representation of a modelling language on medium. The medium itself can be restricted to a specific form, such as graphical, textual, or video, but the notation in general can also be restricted in terms of fonts, icons and layout rules. Concrete syntax defines symbols (according to the medium) and rules for their combination, as well as their correspondence to the abstract syntax of the language. The role of notation in modelling is thorougly discussed in [37].

The **semantics** of a modelling language deals with its meaning. It is conventionally defined in terms of a *semantic domain* and a *semantic mapping* [24]. According to [24], the semantic domain captures the "*decisions about the kinds of things language should express*" [24, p. 68], while the semantic mapping establishes the correspondence from syntactic elements to the semantic domain.

It is often considered that the abstract syntax of the modelling language does not deal with semantics [24]. Nonetheless, the metamodel of the modelling language actually represents a particular **conceptual foundation** of the language, i.e. a specific classification of concepts to be used in discourse about the "world" [18]. The metamodel thus provides a particular *ontological position*, as it filters the view on the "world" one chooses within the modelling language [18]. It is even argued in [18] that all other aspects of the modelling language depend on its conceptual foundation. Thus, the conceptual foundation is an important (if not crucial) aspect of the modelling language semantics. It may be even argued, based on the research in the area of linguistics (e.g. [13,23]) that syntax and semantics of a language are not that clear-cut.

In the following, we will argue that this traditional approach to modelling language definition, and in particular standardisation, needs to be complemented with another dimension, that of **modelling pragmatics** [49]. Pragmatics is concerned with the use of language signs by the user, in the context/situation in which and the purposes for which they are used, as well as meaning and effect they have in their context of use [49,1]. Though it is not widely studied [49], we will show that inadequate consideration of pragmatic aspects of modelling, when defining a modelling language, may affect its capacity to effectively perform its function in modelling. We will equally show that an explicit consideration of pragmatic aspects may provide valuable insights for the scoping and a the design of a better-suited modelling languages.

4.3 Function of the Modelling Language

Modelling language can be regarded as having a *twofold function* in modelling:

1. *Representational function*, the function of representation system for expressing models, in particular for their mechanical manipulation, and

[3] The advantages or disadvantages of using metamodelling for representing abstract syntax are discussed in e.g. [24].

2. *Linguistic function*, the function of a natural language to be used in conceptualising and communicating about some domain in a particular modelling situation.

This twofold functioning puts different and often conflicting requirements on the modelling language definition.

Representational Function. In its representational function, a modelling language should accommodate the formulation of models, while allowing their mechanical manipulation. For this purpose, the representation system should *at least* have a well defined *abstract* syntax *prior* to developing tools that implement model manipulations. To reuse the implemented manipulations, the representation system is also typically required to remain *fixed* once defined, i.e. not evolving dynamically in different situations of use.

The potential added value of the modelling language, from this perspective, lies primarily in 1) the re-usability of the representation system and of the associated manipulations across different modelling problems, and 2) the extent to which the language specification is machine readable.

The re-usability of a language relates to its *expressiveness* [18], i.e. to how many different (conceptions of) domains a modelling language allows to model. It can largely be influenced *at language design time*, when its designers identify and restrict the *intended set of models* expressible by the representation system. This is done through the choices relative to the (levels of genericity of the) *conceptual foundation* and to the *syntactic-semantic restrictions* incorporated into the language definition.

The requirement of machine readability is driven by the need for the automated manipulation of models. For representations produced in the modelling language to be precisely interpreted (by machines), a *formal*, i.e. precise and unambiguous, definition of both abstract syntax and semantics of the representation system, usually in a mathematical language, is required. In this context, modelling language is seen as a formal language, and its definition is of normative character.

However, the formal perspective on modelling languages focuses on purely referential aspects of meaning [13], and excludes all context-dependent aspects of meaning in language [1,49], which fall in the area of pragmatics. In particular, this perspective on modelling languages disregards the potential influence of 'labels' from natural language (as part of language's conceptual foundation) on the understanding of the modelling language by an observer, as well as on its use in the process of domain conceptualisation in the given modelling situation. Clearly, from this perspective, a modelling language does not function as a 'human language', but only as an ultimately syntactic carrier of models.

Linguistic Function. In its linguistic function, a modelling language should provide *language* to support the activities of conceptualisation and communication taking place in a modelling situation [27,42].

From this perspective, the potential added value of a modelling language consists in its capacity to 1) frame the discourse about the domain in a modelling situation, and to 2) facilitate shaping and expressing the conception of a domain formed/agreed by an observer in a model that is fit for its purpose. These dimensions combined provide what is usually referred to as the *suitability* or *utility* [42] of a modelling language.

Whether the given modelling language is suitable for some modelling problems is largely contextual and cannot be fully determined *a priori*. The pragmatic aspects of a modelling situation in which the given modelling language is *actually used* determine the degree of its suitability. This point is developed in the following discussion.

As defined in Section 3, a modelling situation is at least characterised by a wider context of modelling, the goals of the situation (including the purpose of the model), and the observer involved in modelling. Additionally, the fixed/standard modelling language to use, **ML**, may also be (and typically is) selected *prior* to engaging in the modelling effort. As illustrated in Figure 2a, the central question, in this context, is to which extent the selected **ML** is capable to effectively support the creation of conception $\mathbf{c}_d$, as well as its externalisation into **m** suitable for the purpose **p** (in terms of representational system provided by **ML**). Let us try to provide a tentative answer to these questions.

As discussed in 4.2, **ML** provides an embedded filter on the "world", i.e. its conceptual foundation. In the modelling situation, this filter is meant to constrain, or at least influence, the way observer **O** forms the conception of a domain $\mathbf{c}_d$. To which extent an 'externally' imposed language **ML** can interact with the observer's process of conceiving a particular domain **d**?

The way in which a particular individual/observer observes and conceptualises the "world" is shaped by the factors of biological, cognitive, cultural, and social (thus also educational and professional) background of the individual [44]. These factors shape the linguistic personality [21] of the individual, which is illustrated

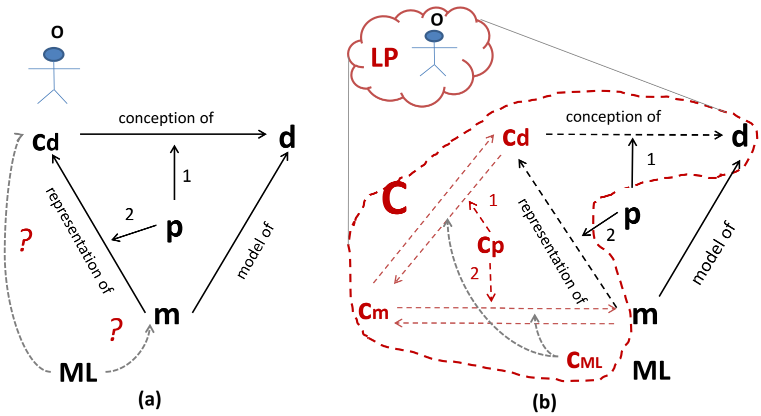


Fig. 2. The role of modelling language in a modelling act

as a space **LP** around observer **O** in Figure 2b. This is also addressed as the individual's world view, or the observer's pre-conceptions in [42]. The **LP** of an observer **O** affects his/her *default interpretation* of concepts embedded in **ML** [50], i.e. his/her *default* $\mathbf{c}_{ML}$. The shaping of $\mathbf{c}_{ML}$ is further influenced by the characteristics of the modelling situation: its wider context, e.g. the particular enterprise and/or project, language(s) spoken in (particular groups within) the enterprise [26], model purpose **p**, other participants (in the case of collaborative modelling) etc. In this context, the understanding of conceptual foundation of **ML** follows the principles of human use of natural languages. In the use of natural languages, the words and/or linguistic utterances are given their precise meaning within the entire context of linguistic communication, and the function linguistic utterances have in the communication context [13,12]. Therefore, the contextualised understanding of **ML**, i.e. *contextualised* $\mathbf{c}_{ML}$, arises by taking into account the entire context in which modelling takes place.

An expert modeller can be expected to have less difficulties in understanding and using a particular **ML**, because of his/her education and consistent experience in modelling. However, it is reasonable to assume that a prototypical stakeholder of enterprise modelling (involved as observer in a modelling situation) requires an increased *mental effort* [5,3] to understand and use **ML** [56]. In fact, the **LP** of a prototypical enterprise modelling stakeholder differs most likely significantly from the specific ontological position, i.e. level of abstraction, embedded in **ML**. The latter typically provides the level of abstraction higher than the one in which most stakeholders are used to reason. Adopting higher than usual levels of abstraction in reasoning also increases mental effort needed for stakeholders in this task [51]. The training and consistent experience in *using* a **ML** by stakeholders may possibly remedy this, but it is unlikely that each enterprise modelling stakeholder will be trained to the modelling languages used in his/her particular context.

If we add to this picture the typical conceptual complexity of standardised **ML** (see Section 2), it can quickly result in *cognitive overload* [5,36] observers, when *only* trying to understand the **ML**. Such a situation rapidly hinders the understanding of other elements in the modelling effort [56], i.e. $\mathbf{c}_d$, $\mathbf{c}_m$ and $\mathbf{c}_p$. This is one of the potential sources of problems with the use of **ML** in the practical modelling situations.

Secondly, **ML** is meant to affect observer's creating and aligning of conceptions $\mathbf{c}_d$ and $\mathbf{c}_m$, as illustrated in Figure 2b. As model purpose **p** at the same time drives this process(see Section 3), the given **ML** should provide sufficient constructs to form $\mathbf{c}_d$, and allow to express (by the means of **ML**'s representation system) the $\mathbf{c}_d$ for the purpose **p** into an artefact **m**. Whether a particular **ML** used in a modelling situation allows to do so is *not* only dependent on the selected **ML**, but primarily on the model purpose **p**, and the observer **O**, as discussed so far.

If, in the given modelling situation, **ML** (i.e. $\mathbf{c}_{ML}$) lacks constructs to express all the relevant aspects of $\mathbf{c}_d$ for a purpose **p**, the natural attitude in the modelling practice is to invent a 'dialect' that provides the best *cognitive fit* [37] for the purpose **p** at hand. As the **ML** is typically a *fixed* language, it is not likely that all the possible relevant concepts and constructs for all the modelling situations could be preconceived in such a language anyway. This suggests that the need to adapt the modelling language in the given modelling situation is likely to be present despite the standardisation/unification efforts.

What these discussions suggest is that, in order to maximise its added value in terms of linguistic function, the **ML** should be as closely aligned as possible with the characteristics of the *actual modelling situation* in which it is used, e.g. in terms of the provided ontological position, vocabulary, coverage of specific aspects, level of detail in modelling, specific form and symbols used, etc. This also means that an ideally suitable language could not be *a priori* fixed, but that the adaptability to specific modelling situations has to be *designed in* the language.

Moreover, when defining a fixed modelling language or when selecting it for a particular modelling effort, one should be careful when assuming that the influence of the observer's **LP** on his/her conceptions can be overcome by imposing a filter from **ML** from which to look at the domain. Our discussions rather suggest that the conceptual foundation of **ML** chosen for a modelling situation should be as close as possible to the way stakeholders would discuss the particular modelling problem, if there would be no restrictions on the language to use whatsoever[4].

This is even more critical for language standardisation efforts. The *standardised* **ML** is typically thought as an effective solution to also standardise the (conceptualisation and) communication about some problem area, regardless of the involved stakeholders, and the purposes for which modelling should be done. The discussions so far rather suggest that, because of differences in individuals' world views, the $\mathbf{c}_{ML}$ is likely to be different for each individual, even in the case of standard **ML**. It indeed suggests that meta-discussions are likely unavoidable, if not even necessary to reach a sufficiently similar $\mathbf{c}_{ML}$ between human actors involved in modelling. Furthermore, it also suggests that the degree of complexity of a modelling language has to be manageable by humans, such that the language indeed facilitates rather than hinders the modelling effort.

5 Pragmatics under the Carpet of Standardisation?

The previous section identifies and discusses the opposing forces that influence the definition of a modelling language. These forces, stemming from its aforementioned functions, evidently need to be carefully balanced so that the modelling language provides the added value when *used* in modelling situations. Based on the discussions in Sections 2 and 4, we can observe that standardisation efforts mainly aim to maximise the potential benefits of the modelling language in its representation function. In these efforts, its linguistic function is very lightly, if at all, considered.

[4] In this context, the notation of **ML** can also play an important role in facilitating understanding and use of **ML**, see [37].

To support this discussion, we will analyse the BPMN standard and its use (as reported in [57,58,43,53,20]). The OMG's BPMN is developed with the primary goal "*to provide a notation that is readily understandable by all business users, from the business analysts that create the initial drafts of the processes, to the technical developers responsible for implementing the technology that will perform those processes, and finally, to the business people who will manage and monitor those processes.[..] In doing so, BPMN will provide a simple means of communicating process information to other business users, process implementers, customers, and suppliers*" [39, p.31].

Business users are identified in the specification document as the primary audience of BPMN models. At the same time, BPMN aims to create standardized bridge between business process design and process implementation, and to enable visualizing process models defined in languages optimized for their execution.

How does the BPMN standard reflect the needs of its primary business users, and how does it support creating models for these 'business-oriented' purposes? According to [58,43], these purposes in practice are mainly process documentation, continuous process management, knowledge management, and organisation redesign. Furthermore, a very small subset of essential BPMN constructs is used for these purposes [58][5]. However, this does not imply that such a small subset is sufficient for the needs of business users. As reported in [53,43], practitioners lack constructs for expressing business rules, organisational resources and roles, risks, performance, etc. [43,53]. This corroborates the pattern of extension of BPMN models observed in [43]: the models are extended with the symbols allowing to capture organisational information, such as data, risks, resources, documents etc. "*This situation points to BPMN being a pure process modeling language. Users, however, often are concerned with enterprise modeling [...] beyond the mere depiction of the control flow of their business operations.*" [43, p.189].

While targeting primarily business users, a closer look at the internal complexity of BPMN standard reveals the language rather geared towards the advanced technical purposes of process modelling, e.g. workflow engineering, process simulation and systems specification etc [43]. However, such an advanced modelling is very rarely applied in practice [43]. Furthermore, a rather significant numbers of constructs of BPMN standard is reported by practitioners as superfluous, e.g. some highly differentiated event constructs, gateways etc [58,43]. This complexity is reported to negatively affect the ease of use of the language [43,20]. If reported as overwhelming, should the language complexity then somehow be managed? Should the constructs used only for advanced process modelling purposes be hidden from other users of the language?

Similar observation of the BPMN complexity motivated the work on defining a 'simplified' version of the language, namely Simple BPMN (SBPMN) [20], targeted at business users of process modelling. The original standard is simplified

[5] We have discussed the usage of BPMN constructs for different purposes in more details in Section 2.2 of the paper.

by excluding or modifying constructs that require technical knowledge or that can cause confusion when used by business users. It does not surprise that the usability tests performed and reported in [20] asses SBPMN as easier to use for business users than BPMN.

Therefore, we can observe that BPMN targets too broad a range of process modelling purposes, as well as very diverse audience. The approach to its standardisation is oriented towards maximising the benefits of *only* the *representation system* across rather technical uses of process models. However, BPMN does not seem to accommodate well the needs of its business audience: it does not come with relevant constructs for modelling processes from the business perspective. Also, as a fairly complex language it is difficult to use by this audience. In our view, the standardisation is done without adequately considering the pragmatic needs (in terms of audience, model purposes, constructs needed to structure the discourse about the domain) of modelling situations in which BPMN is/should be used. As we have seen, this affects its suitability, i.e. how it performs the linguistic function. The examples of SBPMN and of BPMN extensions covering organisational aspects illustrate the strategies to overcome the lack of BPMN's suitability. Such 'variants' seem to have better chances of cognitive fit for 'business-oriented' purposes of process modelling, and they certainly demonstrate better cognitive effectiveness than the full-blown standard.

More importantly, this raises the question whether such 'simplified variants', tuned to particular purposes/audiences, should be part of a standard process modelling language, too. Could the BPMN standard fulfil its stated objectives effectively, if it does not include these 'variants'? Similar findings result also from the study of practical use and support needed by architectural languages [35].

6 Conclusion and Outlook

Instead of sweeping pragmatics under the carpet, how should the pragmatic needs for language support, as well as language use data, stand in relation with the standard? Should the needs of language users be the driving force in creating and scoping the (standard) language?

To shed more light on this question, this paper discussed the role of the modelling language from a fundamental perspective. We identified its twofold role in modelling, and argued that it has to be carefully considered and balanced when defining the (standard) modelling language. In other words, we argued that it makes sense to explicitly consider the pragmatic needs when defining, revising or evolving a fixed modelling language.

What is the general competence area of a modelling language, e.g. process modelling language? Should the process modelling language cover only a single, process, aspect or include aspects such as business rules, resources, and similar aspects related to organisational modelling, which are relevant for purposes such as process documentation, process monitoring and improvement, and re-engineering? The same questions can be asked for any modelling language.

Our research suggests that the scope of a fixed modelling language should be in line with the context in which it will be used, i.e. in line with purpose(s) for which models are produced using that language. If enough support is not a priori provided in the language, purpose-specific 'variants' will emerge to compensate for the missing suitability.

If the fixed modelling language aims to cover too many purposes at the same time, this is more challenging, as the language tends to become overly complex if there is a strive to accommodate all the different pragmatic needs. Nonetheless, these needs cannot be swept under the carpet, as they will reappear when the language is used in the actual modelling situations. Should then the standard language for e.g. process modelling be reorganised into multiple (modular) languages within a standard for process modelling? Does this suggest a different approach towards the standardisation of modelling languages?

We believe this is one of the promising directions to explore in the future research. The modular organisation of languages could have language 'chunks' scoped and geared towards purpose(s) for which the 'chunk' is used. The *right* language for the modelling situation at hand could then be woven out of the different 'chunks', based on the pragmatic needs of the actual modelling situation. This also suggests that the pragmatic aspects of the 'chunks' need to be made as explicit as it is possible. Such a modular organisation could also decrease the cognitive load needed for the understanding, selecting and using the language for chosen modelling purposes.

In addition, the needs for *situational adaptability* and *evolution* of the language will always be present, as it is not feasible to pre-define all the possible circumstances in potential modelling situations, nor to predict the evolution of the 'reality' for whose modelling the language support will be needed. The strategy of including all these missing aspects within a *single* language would make it at some point overly complex and virtually unusable. In our view, this is another argument in favour of modular organisation of modelling languages.

Of course, this calls for more research on instruments of modelling language creation, adaptation and combination, as well as more research on the related tool mechanisms supporting this. Indeed, there is a growing research interest in modelling language and tool flexibility [32,11,54]. Our research aims to contribute to this stream of research by providing a conceptual framework from which to understand the role and, in particular, the use of the modelling language. This paper argued that the added value of a modelling language cannot be evaluated without considering its use, i.e. pragmatics dimension. Our belief is that only by identifying the factors determining the added value of the language, we can be in the position to make grounded decisions on optimal scope and design of the language.

Acknowledgements. This work has been partially sponsored by the *Fonds National de la Recherche Luxembourg* (`www.fnr.lu`), via the PEARL programme.

References

1. Allwood, J.: Some Remarks On the Relationship Between the Semantic and the Pragmatic Web. In: ICPW, pp. 35–39 (2008)
2. Anaby-Tavor, A., Amid, D., Fisher, A., Bercovici, A., Ossher, H., Callery, M., Desmond, M., Krasikov, S., Simmonds, I.: Insights into enterprise conceptual modeling. Data Knowl. Eng. 69(12), 1302–1318 (2010)
3. Baddely, A.: Working memory: Theories, models and controversies. Annual Review of Psychology 63, 1–29 (2012)
4. Band, I., Bjeković, M., Chaney, C., Lindokken, K., van Dis, E.: Modeling the Insurance Enterprise. Technical Report W12B, The Open Group (2012)
5. Bannert, M.: Managing cognitive load-recent trends in cognitive load theory. Learning and Instruction 12, 139–146 (2002)
6. Bjeković, M., Proper, H.A., Sottet, J.S.: Towards a coherent enterprise modelling landscape. In: PoEM (Short Papers), CEUR-WS.org (2012)
7. Bjeković, M., Sottet, J.S., Favre, J.M., Proper, H.: Framework for Natural Enterprise Modelling. In: IEEE CBI Conference (2013) (to appear)
8. van Bommel, P., Hoppenbrouwers, S., Proper, H., Roelofs, J.: Concepts and strategies for quality of modeling. In: Innovations in Information Systems Modeling. IGI Publishing (2008)
9. Bubenko Jr., J.A., Persson, A., Stirna, J.: An Intentional Perspective on Enterprise Modeling. In: Salinesi, C., Nurcan, S., Souveyet, C., Ralyté, J. (eds.) Intentional Perspectives on Information Systems Engineering, pp. 215–237. Springer (2010)
10. Chiprianov, V., Alloush, I., Kermarrec, Y., Rouvrais, S.: Telecommunications Service Creation: Towards Extensions for Enterprise Architecture Modeling Languages. In: ICSOFT (1), pp. 23–28 (2011)
11. Cho, H., Sun, Y., Gray, J., White, J.: Key challenges for modeling language creation by demonstration. In: ICSE FlexiTools Workshop (2011)
12. Clark, H.: Arenas of Language Use. University of Chicago Press (1993)
13. Cruse, A.: Meaning in Language: An Introduction to Semantics and Pragmatics. Oxford Textbooks in Linguistics. Oxford University Press (2011)
14. Davies, I., Green, P.F., Rosemann, M., Indulska, M., Gallo, S.: How do practitioners use conceptual modeling in practice? Data Knowl. Eng. 58(3), 358–380 (2006)
15. de Kinderen, S., Gaaloul, K., Proper, H.A.E.: Integrating Value Modelling into ArchiMate. In: Snene, M. (ed.) IESS 2012. LNBIP, vol. 103, pp. 125–139. Springer, Heidelberg (2012)
16. Elahi, G., Yu, E.S.K., Annosi, M.C.: Modeling Knowledge Transfer in a Software Maintenance Organization - An Experience Report and Critical Analysis. In: Stirna, J., Persson, A. (eds.) PoEM 2008. LNBIP, pp. 15–29. Springer, Heidelberg (2008)
17. Fabro, M.D.D., Valduriez, P.: Towards the efficient development of model transformations using model weaving and matching transformations. Software and System Modeling 8(3), 305–324 (2009)
18. Falkenberg, E.D., Hesse, W., Lindgreen, P., Nilsson, B.E., Oei, J., Rolland, C., Stamper, R., Van Assche, F., Verrijn-Stuart, A., Voss, K.: FRISCO - A Framework of Information System Concepts - The FRISCO Report. Technical report, IFIP WG 8.1 Task Group FRISCO (1998)
19. Feltus, C., Dubois, E., Proper, H.A., Band, I., Petit, M.: Enhancing the ArchiMate® standard with a responsibility modeling language for access rights management. In: SIN, pp. 12–19 (2012)

20. Fernández, H.F., Palacios-González, E., García-Díaz, V., Pelayo, G., Bustelo, B.C., Sanjuán Martínez, O., Cueva Lovelle, J.M.: SBPMN – An easier business process modeling notation for business users. Computer Standards & Interfaces 32(1), 18–28 (2010)
21. Fesenko, O.P.: Structure of Linguistic Personality in the Aspect of the Center-Periphery Theory. Middle-East Journal of Scientific Research 16(3), 397–401 (2013)
22. Frank, U.: Some Guidelines for the Conception of Domain-Specific Modelling Languages. In: EMISA, pp. 93–106 (2011)
23. Geeraerts, D., Cuyckens, H. (eds.): The Oxford Handbook of Cognitive Linguistics. Oxford University Press (2010)
24. Harel, D., Rumpe, B.: Meaningful Modeling: What's the Semantics of "Semantics"? IEEE Computer 37(10), 64–72 (2004)
25. Hoppenbrouwers, S.J.B.A., Wilmont, I.: Focused Conceptualisation: Framing Questioning and Answering in Model-Oriented Dialogue Games. In: van Bommel, P., Hoppenbrouwers, S., Overbeek, S., Proper, E., Barjis, J. (eds.) PoEM 2010. LNBIP, vol. 68, pp. 190–204. Springer, Heidelberg (2010)
26. Hoppenbrouwers, S.J.B.A.: Freezing Language; Conceptualisation processes in ICT supported organisations. PhD thesis, University of Nijmegen, Nijmegen, The Netherlands (2003)
27. Hoppenbrouwers, S.J.B.A., Proper, H.A., van Reijswoud, V.: Navigating the Methodology Jungle – The communicative role of modelling techniques in information system development. Computing Letters 1(3) (2005)
28. Hoppenbrouwers, S.J.B.A., Proper, H.A., van der Weide, T.P.: A Fundamental View on the Process of Conceptual Modeling. In: Delcambre, L.M.L., Kop, C., Mayr, H.C., Mylopoulos, J., Pastor, Ó. (eds.) ER 2005. LNCS, vol. 3716, pp. 128–143. Springer, Heidelberg (2005)
29. Hoppenbrouwers, S.J.B.A., Proper, H.A., van der Weide, T.P.: Understanding the Requirements on Modelling Techniques. In: Pastor, Ó., Falcão e Cunha, J. (eds.) CAiSE 2005. LNCS, vol. 3520, pp. 262–276. Springer, Heidelberg (2005)
30. Iacob, M.E., Jonkers, H., Lankhorst, M., Proper, H.A., Quartel, D.: ArchiMate 2.0 Specification. The Open Group (2012)
31. Karlsen, A.: Enterprise Modeling Practice in ICT-Enabled Process Change. In: Johannesson, P., Krogstie, J., Opdahl, A.L. (eds.) PoEM 2011. LNBIP, vol. 92, pp. 208–222. Springer, Heidelberg (2011)
32. Kimelman, D., Hirschman, K.: A Spectrum of Flexibility-Lowering Barriers to Modeling Tool Adoption. In: ICSE FlexiTools Workshop (2011)
33. Kort, C., Gordjin, J.: Modeling Strategic Partnerships Using the E3value Ontology: A Field Study in the Banking Industry. In: Handbook of Ontologies for Business Interaction, pp. 310–325. IGI Global (2008)
34. Krogstie, J., Sindre, G., Jørgensen, H.D.: Process models representing knowledge for action: A revised quality framework. EJIS 15(1), 91–102 (2006)
35. Malavolta, I., Lago, P., Muccini, H., Pelliccione, P., Tang, A.: What Industry Needs from Architectural Languages: A Survey. IEEE Trans. Software Eng. 39(6), 869–891 (2013)
36. Moody, D.L.: Cognitive load effects on end user understanding of conceptual models: An experimental analysis. In: Benczúr, A.A., Demetrovics, J., Gottlob, G. (eds.) ADBIS 2004. LNCS, vol. 3255, pp. 129–143. Springer, Heidelberg (2004)
37. Moody, D.: The "Physics" of Notations: Toward a Scientific Basis for Constructing Visual Notations in Software Engineering. IEEE Transactions on Software Engineering 35(6), 756–779 (2009)

38. OMG: UML 2.0 Superstructure Specification – Final Adopted Specification. Technical Report ptc/03–08–02 (August 2003)
39. OMG: Business Process Model and Notation (BPMN), Version 2.0 (January 2011)
40. OMG: OMG Systems Modeling Language (OMG SysML), Version 1.3 (June 2012)
41. Opdahl, A., Berio, G., Harzallah, M., Matulevicius, R.: Ontology for Enterprise and Information Systems Modelling. Applied Ontology 7(1), 49–92 (2012)
42. Proper, H.A., Verrijn-Stuart, A.A., Hoppenbrouwers, S.: On Utility-based Selection of Architecture-Modelling Concepts. In: APCCM 2005, pp. 25–34 (2005)
43. Recker, J.: Opportunities and constraints: the current struggle with BPMN. Business Proc. Manag. Journal 16(1), 181–201 (2010)
44. Rohrer, T.: Embodiment and Experientialism. In: Geeraerts, D., Cuyckens, H. (eds.) The Oxford Handbook of Cognitive Linguistics, pp. 25–47. Oxford University Press (2010)
45. Rothenberg, J.: The Nature of Modeling. In: Artificial intelligence, Simulation & Modeling, pp. 75–92. John Wiley & Sons, Inc., USA (1989)
46. Stachowiak, H.: Allgemeine Modelltheorie. Springer, Germany (1973)
47. Stamper, R., Liu, K., Hafkamp, M., Ades, Y.: Understanding the roles of signs and norms in organizations - A semiotic approach to information systems design. Behaviour & Information Technology 19(1), 15–27 (2000)
48. Thalheim, B.: The Theory of Conceptual Models, the Theory of Conceptual Modelling and Foundations of Conceptual Modelling. In: Handbook of Conceptual Modeling, pp. 543–577. Springer (2011)
49. Thalheim, B.: Syntax, Semantics and Pragmatics of Conceptual Modelling. In: Bouma, G., Ittoo, A., Métais, E., Wortmann, H. (eds.) NLDB 2012. LNCS, vol. 7337, pp. 1–10. Springer, Heidelberg (2012)
50. van der Linden, D.J.T., Hoppenbrouwers, S., Lartseva, A., Molnar, W.: Beyond terminologies: Using psychometrics to validate shared ontologies. Applied Ontology 7(4), 471–487 (2012)
51. Wilmont, I., Hengeveld, S., Barendsen, E., Hoppenbrouwers, S.: Cognitive mechanisms of conceptual modelling - how do people do it? In: Ng, W., Storey, V.C., Trujillo, J.C. (eds.) ER 2013. LNCS, vol. 8217, pp. 74–87. Springer, Heidelberg (2013)
52. Winograd, T., Flores, F.: Understanding Computers and Cognition - A New Foundation for Design. Ablex Publishing Corporation (1986)
53. Wohed, P., van der Aalst, W.M.P., Dumas, M., ter Hofstede, A.H.M., Russell, N.: On the Suitability of BPMN for Business Process Modelling. In: Dustdar, S., Fiadeiro, J.L., Sheth, A.P. (eds.) BPM 2006. LNCS, vol. 4102, pp. 161–176. Springer, Heidelberg (2006)
54. Zarwin, Z., Bjeković, M., Favre, J.M., Sottet, J.S., Proper, H.: Natural modelling. Journal of Object Technology (to appear, 2014)
55. Zivkovic, S., Kühn, H., Karagiannis, D.: Facilitate Modelling Using Method Integration: An Approach Using Mappings and Integration Rules. In: ECIS, pp. 2038–2049 (2007)
56. Zugal, S., Pinggera, J., Reijers, H., Reichert, M., Weber, B.: Making the case for measuring mental effort. In: Proceedings of the Second Edition of the International Workshop on Experiences and Empirical Studies in Software Modelling, p. 6. ACM (2012)
57. zur Muehlen, M., Ho, D.T.: Service Process Innovation: A Case Study of BPMN in Practice. In: HICSS, p. 372 (2008)
58. Muehlen, M.z., Recker, J.: How Much Language Is Enough? Theoretical and Practical Use of the Business Process Modeling Notation. In: Bellahsène, Z., Léonard, M. (eds.) CAiSE 2008. LNCS, vol. 5074, pp. 465–479. Springer, Heidelberg (2008)

Overview of Business Process Modeling Languages Supporting Enterprise Collaboration

Hodjat Soleimani Malekan and Hamideh Afsarmanesh

Informatics Institute, University of Amsterdam, P.O. Box 94323, 1090 GH Amsterdam, The Netherlands
{h.soleimanimalekan,h.Afsarmanesh}@uva.nl

Abstract. Enterprises endeavor to provide innovative services and competitive advantages, by constituting Collaborative Networks (CNs). Each enterprise performs a set of Business Processes (BPs), and through developing integrated BPs in CNs, enterprises can jointly produce stronger capabilities. However, selecting appropriate BP Modeling Languages (BPMLs) for the purpose of formalizing BPs in the CNs are challenging. In surveys published so far, mostly some general features of the main BPMLs are contrasted. But these comparisons analyze neither the features and peculiarities identifying different categories of BPMLs, nor the requirements to be fulfilled in CNs. This article introduces a systematic evaluation method, which first proposes a new categorization for BPMLs. Then, focusing on enterprise collaboration requirements, a specific set of Key Requirement Indicators (KRIs) is identified for CNs, related to what they require from BPMLs. Finally, the introduced BPML categories are discussed against the KRIs, in order to assess their suitability in supporting CNs.

Keywords: Business Processes, Collaborative Network, Business Process Modeling Languages, Virtual Organizations (VO), VO Breeding Environments, Service-Oriented Architecture.

1 1 Introduction

Adopting Business Process (BP) modeling technologies, including the introduced languages, standards, tools, and techniques, have greatly influenced enterprises toward capturing opportunities, reducing costs, and increasing productivity.
The BP technologies themselves however, have also been affected by high demand of market, as well as the step-wise maturity of Business Process Management (BPM) theories. This has caused rapid changes in the last decade in developed BP Modeling Languages (BPMLs), tools, and standards, while also creating challenges for selecting and adopting suitable BPMLs in networked enterprises [3].

Enterprises collaborate with each other in different forms to achieve their common goals [1]. Collaboration in networks of enterprises aims at aligning all available capacities and capabilities towards providing value-added-services [29]. Service Oriented Architecture (SOA) is an expedient paradigm to setup such collaboration. In SOA, services are designed to realize the capabilities of enterprises from their

B. Shishkov (Ed.): BMSD 2013, LNBIP 173, pp. 24–45, 2014.

resources [23], while BPs are required to describe the execution of services [24]. Therefore, creating and sharing formalized definition of BPs constitute some main challenges in accomplishment of SOA-based networked enterprises. This means that adopting more expressive and unambiguous BP technology can result in realizing more successful co-creation in Collaborative Networks (CNs). For this purpose, besides the challenge of selecting generally strong and expressive BPMLs amongst different technologies, evaluation of peculiar CN-related requirements is also needed.

Further to the purpose of BP modeling, the BPMLs differ from each other in their approach to design, analysis, and enacting of the BPs. Several published surveys on BPMLs e.g. [16], [18], and [26] have already tackled the comparison between certain features of the main BPMLs. Most contemporary surveys focus on comparing a few BPML standards and tools (e.g. BPMN vs. BPEL). However, there is a lack of emphasis on comparing at higher level, different categories of BPMLs with each other, to which the specific standards or tools may belong.

From a top-down point of view, each category includes a set of BPMLs with many (if not all) common characteristics, such as the graphical depiction, formal representation, etc. Such categorization and when identifying the most suitable category for CNs, can on one hand reduce and thus facilitate the number of alternative BPMLs among which the enterprises can select. On the other hand, all BPMLs that fall under the most suitable category for CNs can be adopted interchangeably. In other words, enterprises can better cope with rapid changes in BPML standards, e.g. outdating a BPML in the most suitable category, may be remedied by using another BPML from the same category.

For the purposes of performing analysis and evaluating BPMLs for their suitability in support of CNs, a novel method is introduced to manifest CN's characteristics and to assess different BPML categories against them. The assessment of peculiarities and specifications of BPML categories are discussed subjectively, from both BP aspects and the CN's point of view. Finally, our results are validated by experts from the field of Collaborative Networks.

In this article, first the main research outline including the definition of main concept and the applied method are presented (in section2). Thereafter, BPML categories and their position in our evaluation roadmap are reviewed and explained (in section3). Founded on collaboration purposes, a number of most relevant criteria for comparing the BPML categories, and analyzing them for the aim of supporting enterprise collaborations is introduced (in section 4). Then, the results of our approach and their validation are discussed (in section5). Finally, our conclusions are presented (in section6).

2 Research Outline

Within the collaborative-networked environment, enterprises have the opportunity to share their resources through collaboration, including their knowledge, information, but also their provided services, which can be best achieved by means of formalized BPs [3]. Furthermore, collaborative BP integration is aimed by enterprises to accomplish value-added business services, beyond the capabilities and capacities of their individual organizations [2].

In this section, after a short review of related definitions for CNs and BPs, our proposed evaluation method is specified for analysing the BPML categories against the CN requirements.

2.1 Background and Definitions

Before focusing on the evaluation process, we provide some specifications from the CN discipline. A general definition of Collaborative Network is presented in [3] as: "*an alliance constituting a variety of entities that are autonomous, geographically distributed, and heterogeneous in terms of their operating environment, decision making, culture and social capital, that cooperate/collaborate to better achieve common/compatible goals, and their interactions are supported by the computer networks.*" The above characteristics shape the enterprises' collaboration structure and behaviour.

CNs appear in different forms. Virtual Organization (VO) and VO Breeding Environment (VBE) are the two main forms of CNs. In a VO, partners choose co-working and sharing of their BPs and other resources to accomplish their common goals. The motivations for such coalition are commonly formed either around specific market targets or certain innovation purposes. VBEs, which establish long-term alliances of organizations, capture and store BPs of partners in their directories. In the context of the VBE, usually a VO broker seizes an opportunity and chooses the participant organizations for the VO. The VO broker also considers selecting and integrating BPs of different organizations to shape the new VO, responding to achievable opportunities [1].

An exemplary scenario for BP integration in the CN could be as follows. In telecommunication industry, a VO broker may aim at providing a new discounted service package, based on the existing/stored services in its telecommunication VBE' directory. The broker develops a workflow to accomplish the discounted package from existing VO's business services (namely: the phone service, internet connection, mobile service, etc.) for that purpose. For composing a workflow for the discounted package however, the broker needs to have access to concise definition of the services provided by the VO members. But at present, VO member organizations provide different BP formalizations, which are also mostly ambiguous and underspecified. Therefore, the challenge is how to improve organizations' formalization and representation of their BPs, as needed in CNs.

BPs are typically defined as a series of one or more linked procedures or activities, which collectively realize a business objective or policy-related goal. Workflow Management System (WFMS) can automate and control the execution of the BPs. The notion of BPM comprises concepts, methods, and techniques to support organizational aspect of processes, which are needed for the design, administration, configuration, enactment, and analysis of BPs [39]. It also covers the "*diagnosis*" aspect of the BPs further to the WFMS lifecycle [35].

BP Modeling aims at representing an abstract but meaningful demonstration of the real business domains. In [12], the design and execution aspects of the BPs are introduced as the main concerns for BP modeling. This goal is achieved through provision of appropriate syntax and semantics in BPMLs, to meet the BP's requirements [18]. The BPs, formalized by BPMLs are regarded as the centre of focus for productive performance of the enterprises in the CNs. The BPMLs have supported IT-based collaborations based on their maturity and development purposes.

2.2 Evolution of BPMLs in Support of Bi-lateral Collaboration

In order to achieve a better understanding of the role of BPMLs in collaboration context, below the chronological development of BPMLs, from the point of view of supporting collaboration, is briefly addressed. In the 80s, the necessity of process-awareness was recognized, beyond the level that was required for development of Management Information Systems (MIS). Thus, besides understanding the flow of operations in MIS, organizations and business domain experts needed to also understand the information aspects of the BPs in MIS [44].

At the beginning of the 90s, the WFMSs, which initially were intended to facilitate automatic transformation of electronic documents, were then introduced as the new tools to enable business analysts in designing and expressing BPs. For the purpose of depicting information exchange among systems, the behavioural concepts (i.e. the sequence and merge) were then introduced in BP modeling [45]. Afterward in the 90s, applying the Business Process Re-engineering as well as embedding the best business practices in the market, vendors were able to integrate and align separate software modules, under the so-called Enterprise Resource Planning (ERP) systems. To support ERPs, the BPMLs have focused on dynamic aspects of the BPs. Nevertheless, the interactions between the designed modules were not so easy to achieve within the ERPs [24].

Responding to the proliferation needs of the integrating legacy systems into customized applications and ERP modules, the Enterprise Application Integration (EAI) [38] were introduced, and have tried to remedy the problem of inefficient BPs' integration. So, interaction-enabling entities (e.g. messages) gained significance. This level of collaboration provided an infrastructure for cooperation of enterprises through resource sharing, while preserving their heterogeneity.

The more maturity in deployment of XML, in the late 90s, resulted in better integration of applications, and changed the co-working intensity of enterprises to an advanced level, called business to business (B2B) [12]. Later on, coordinating the BPs adopted by companies, concluded in integrating autonomous and independent applications, via loosely-coupled mechanism of SOA [17]. SOA approach tries to establish orchestration and choreography of web services, to achieve their successful cooperation.

Nowadays, BP related topics e.g. the BP mining [37] and diagnosis approaches [35] that address BP monitoring and their continuous improvement, constitute promising research lines.

2.3 Evaluation Roadmap

The expression of the applied method is an important part of every research. An extension of the epistemological model for evaluating the modelling languages introduced in [7], is what we have developed for our purpose. The steps, functions, and outputs of each level are illustrated in Fig.1. In this revised model, which is primarily developed for empirical research evaluation of the BPMLs, we first "*focus*" on defining a clear outline for the evaluation domain, and introduce the BPMLs categorization (addressed in sections 3.1 and 3.2). The concern of the "*contextual model*" step is in positioning the BPs in the CN context, in order to avoid both the misrepresentation of the BP's notion and the CN's discipline. The ARCON model of collaborative networks is applied [3] here for clear positioning of BPs in CNs (addressed in section 4.1).

During the "*Review*" step, the main instances for each BPML category are identified, and an abstract definition of each considered BPML is provided (addressed in section 3.3). The two assessment levels: "*primary guidelines*" and "*secondary guidelines*" respectively specify the Critical Success Factors (CSFs) and the Key Requirement Indicators (KRI) related to BPs' suitability criteria in supporting CNs (addressed in sections 4.2 and 4.3 respectively). The Critical Success Analysis method is then adopted for completion of these steps.

In the "*Context Analysis*" step, analysis of BPMLs in relation to the selected suitability criteria is discussed (addressed in section 5). The final comparison results of BPML categories are presented and validated at the last stage (addressed in Section 6). All mentioned methods and related results are discussed further in the article.

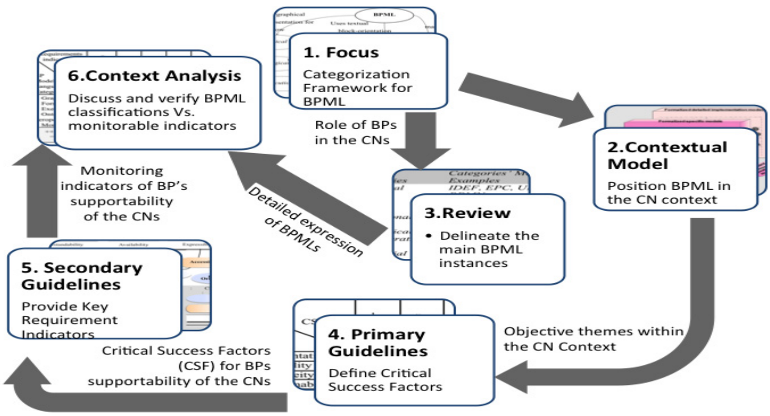


Fig. 1. Our BPML evaluation method – based on approach introduced in [7]

3 BPMLs Put into Context

Aiming to cover various BP modeling tools and standards, which are introduced in the main related publications, and according to the step 1 of our evaluation method in Fig.1, we address specific set of attributes and specifications of the BPMLs for their categorization. Our categories basically focus on recognizing the BPML's capabilities as well as the suitability features in each category, in support of criteria for collaboration. Therefore, first the related literature are considered, and we have classified the BPML-categorization related publications into two classes of: "*General Review*", and "*Particular Evaluation*". In this section, first this classification of BPML-categorization related publications is described, and then for each class, the results of their identified surveys are summarized. We finally introduce six specific categories of BPMLs, which are then uses as the base for further analysis in this article.

3.1 Structuring Related Work

As mentioned earlier, we divide the contemporary reviews of BPMLs into two main classes of "*General Reviews*" and "*Particular Evaluation*" (Fig.2). "General Reviews" are mostly focused on general uses of the BPMLs and encompassing the main specifications of the BPML categories. For instance the work in [12] that focuses on *presenting good BP Modeling Architecture*, where it first addresses aspects of BP modeling applications (i.e. design, run, monitor, etc.), and then introduces the four categories of languages, including: *notation* languages (e.g. BPMN), *execution* languages (e.g. BPEL), *choreography* languages (e.g. WS-CDL), and *process administration* languages (e.g. BPQI). Also, other classifications such as those presented in [16] and [22] fall in this category.

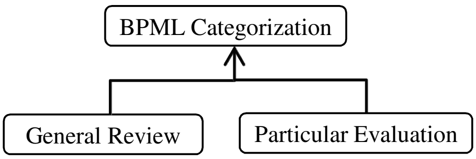


Fig. 2. Structure of reviewed related works on BPML categorization

But, there are other kinds of survey publications, which we classify in the "Particular Evaluations" class. These focus on BPML categorization for specific purposes. The works presented in [8], [18] and [26] are instances in this category. For example, in [8] the BPML categories are introduced around the subject of "introducing an ontological approach for BP modeling", and the suggested categories there include: Descriptive (e.g. BPMN), Procedural (e.g. XPDL), Formal (e.g. PSL), and Ontology-based (e.g. OWL-s).

3.2 BPML Categorization

With the final aim to identify our own set of BPML categories, we have considered publications that appear in scientific databases, and systematically searched and identified the main publications addressing "business process modelling languages" and "business process modelling approaches". Then the minimal set of relevant and related work is selected, and classified either as "general review" or "particular evaluation". A summary of our findings is reported in Table1. This tabular illustration outlines: the main purpose of the author(s), while indicating the class (G: general review, and P: particular evaluation); the conceptual base indicating the main logical perspective applied for their purposes; their addressed/introduced categories; and examples for each category's members.

Table 1. Related works on BPML-Categorization

BY	Main purpose	Conceptual base	Addressed Categories	Categories' Members Examples
Curtis [5]	Overview completeness of BP modeling for software development (P)	Defining an evaluation based on four BP modeling perspectives: "Functional", "Behavioral", "Organizational", "informational"	Procedural Programming	APPL/A
			System Analysis and Design	STATEMATE
			AI languages and approaches	MARVEL
			Formal languages	Context-free grammar
			Data modeling	PMDB
			Functional Languages	HFSP
Aguilar-Saven [9]	Through BPMLs survey presenting a framework for helping users in right BPMLs selection according to their needs (P)	Outlining a two – dimension matrix as the classification framework for BPMLs in axes: "*purpose of the model*" and "*change permissiveness"*	Purpose of model sub-axes: Descriptive for learning/support for process develop and design/ Support for process execution/ Support for IT enactment)	IDEF3 located in "*descripted learning purpose*" and "*active permissiveness*" conjunctions. Gant chart is located in conjunction of "*support for process execution purpose*" and "*passive permissive*" within this grid framework.
			Change permissiveness sub-axes: Active (dynamic models)/ Passive (difficult to interact with)	
Havey [12]	Presenting a Good BP Modeling Architecture (G)	Understanding aspects of BP modeling application's in : "design", "run", "monitor"; and Human and system interactions	*Notation languages*	BPMN, UML
			Execution language	BPEL, WSFL
			Administration and monitoring/ Human interaction/System Interaction	BPRI, BPQL
			Choreography	WS-CDL, WSCI
Mending [34]	Specify a meta-model for XML exchange format (P)	Defining a set of metamodel concepts namely: roles, events, etc. for evaluating BPMLs	Composition	A comparative analysis of series of BP modeling Language has performed, versus metamodeling concepts.
			Choreography	
			Business Analysis	
			Formal Analysis	
Roser [26]	Introduce a categorization used in modeling collaborative BPs (P)	Represent a framework for developing collaborative Processes based on Model Driven Architecture (MDA)	level of abstraction	CIM (BPMN); PIM (BPDM); PSM (WSBPEL)
			Modeling of BP	Private, Public and Collaborative Processes; respectively: BPMN, BPEL and ebXML.
			Notation	Graphical(ARIS); Textual (CDL)
			Standardization	ISO, OMG, W3C
			Tool-support	Tool sets: ARIS-toolset
Lu [18]	Survey on BP Approaches (P)	Address a comparison based on a set of five criteria.	Graph-based Languages	YAWL
			Rule-based languages	Object-Rule-Role approach
De Nicola [8]	Introducing an ontological Approach to BP (modelling (P)	Follow the Idea of "*Inject*" formal semantics in BP modeling tools	Descriptive	BPMN, IDEF
			Procedural	BPEL, XPDL
			Formal	PSL, Petri-Net
			Ontology based	OWL-S, WISMO

Vergidis [36]	Business Processes capabilities in Reengineering (P)	Covering a complete set of BP techniques' range in analyzing and optimizing (some technics are intersected)	Diagrammatic Models	RAD, IDEF/ UML, BPMN, YAWL/ PetriNet
			Mathematical Models	PetriNet
			Business Process Language	BPEL/UML, BPMN, YAWL
Ko [16]	Business Process Standards Survey (G)	BPM Life Cycle (Design, Configuration, Enactment, Diagnosis)	Graphical standards	BPMN, UML AD
			Execution standards	BPEL, YAWL
			Interchange Standards	BPDM, XPDL
			Diagnosis standards	BPQL
			B2B exchange standards	ebXML and RosettaNet.
Melao [40]	E-Business Process Modeling Standards (P)	e-BPM standards in E-business Context	Graphical Notation Standards	BPMN, UML
			Industry-level Standards	Rosettanet
			Cross-industry Standards	BPSS, BPEL
Mili [22]	Business Process Modeling overview (G)	Addressing the business goals and aspects (Functional, Informational, Organizational and Dynamic)	Traditional BP modeling	IDEF, EPC
			Object-oriented BP modeling	UML, EDOC
			Dynamic BP modeling	BPMN, WS-BPEL
			Process Integration BP modeling	ebXML, WS-CDL

Using the results gathered in our related work study of both "general review" and "particular evaluation", and with an eye toward CN expectations from BP formalization, we propose six specific categories for BPMLs, including: "graphical", "formal", "executional, "ontological", "inter-operational", and "monitorial".

These six categories provide full coverage on all peculiarities and characteristics of the BPMLs that we have summarized in Table1. The meta-process that we have adopted for our categorization method, is also briefly depicted in Fig.3.

The main characteristics which we have considered for these six categories also include aspects related to the expected role of BPML in CNs, e.g. providing concise/formal specification, inter-organizational support, etc. Later on, in Section 6.2, we address the evaluation/validation of our results by the CN experts.

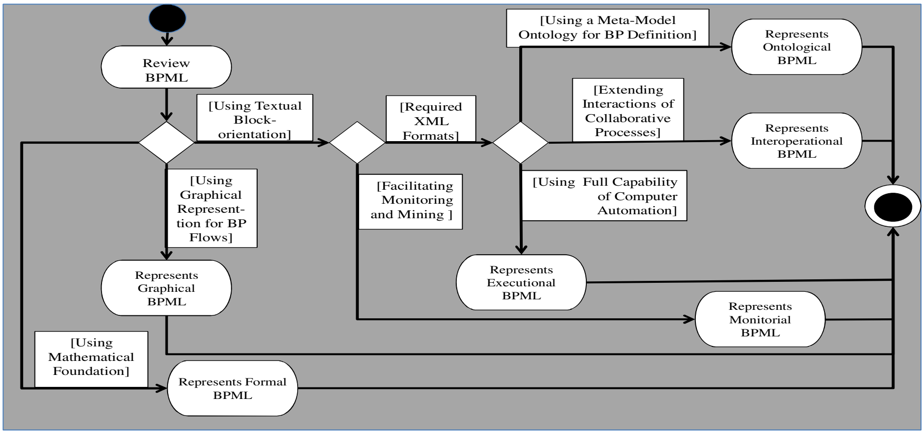


Fig. 3. Meta-Process Method for BPML Categorization

3.3 Delineation of Our BPML Categories

In this section, and according to the step 3 of our evaluation method in Fig.1, the six BPML categories are characterized and their main representative and popular example BPMLs and standards are mentioned. The characteristics identifying each BPML category can then be used by the CN members for the purpose of adopting and utilizing their suitable language. Although in principle it is possible for a BPML to be categorized under more than one of these categories, we have here placed each BPML in its most representative category only.

Graphical BP Modeling Languages.
Rooted in graphical picturesque format, this classical generation of languages has appeared. BP modeling languages in this category mostly emphasize on illustrating the system behaviour and its abstraction. These BPMLs are not typically formal. Example languages in the category include: IDEF, EPC, UML 2.0, and BPMN.

Formal BP Modeling Languages.
Formalization in this category is primarily founded upon mathematical principles. Although, adoption of graphical symbols is common in some of these languages, but difficulties in typical user's understand-ability hold these languages mostly at the theoretical level and for academic utilizations. Example languages include: Petri-Net, Pi-calculus, PSL, and Reo.

Executional BP Modeling Languages.
The idea of automatic execution of BPs by software engines is the base for the formation of this category. The XML structure plays a major role in deployment of this category of languages, and clarifies BPs by their computerized semantics. Additionally, the popularity of BP modeling and specifically the needed support for service invocation in industries are important reasons for development of these languages. Example BPMLs include: BPEL, WS-CDL, XPDL, and YAWL.

Ontological BP Modeling Languages.
To support the modeling requirements set by ontologies, which study and describe the nature of beings and their inter-relations, this category of languages focuses on semantic capture. These languages aims to provide the base and to propose different meta-models that can support an increasing number of other BP modeling languages. The ontological layer in these languages clarifies the roles, entities, and interactions. This category of languages also takes advantage of using XML. Example languages in the category include: OWL-s, WSMO, and BPDM.

Interoperational BP Modeling Languages.
Rooted in business-to-business interaction, this category focuses on modeling public sharable processes in collaborations among many business partners. To accomplish this key concern in inter-operational category, the XML standards are elaborated as the main enablers. Example BPMLs in this category include: RossettaNet and eb-XML/BPSS.

Monitorial BP Modeling Languages.
As discussed previously in section 2, modern business process modeling tends to also address the diagnosis iteration of the BP Lifecycle. Focusing on the Business Activity Monitoring (BAM) point of view, the emphasis is on monitoring and resolving the deadlocks or problems in the flow of the BPs. Furthermore, an unambiguous approach for recognizing/extraction of modeled BPs, based on the dynamic logging of process behaviour, as also addressed in process mining is promising [37]. Example languages in this category include: BPRI and BPQI.

4 CN Requirements Put into Context

Following the roadmap described in 2.3, this section focuses on steps 2, 4 and 5 of our evaluation method in Fig.1, aiming to provide descriptive and concise details about the framework for evaluation of BPML categories against the CN requirements.

Putting emphasis on categories and not every BPML as addressed earlier, we face the multi-aspect evaluation of each phenomenon that requires a methodology to support maximal coverage of the target area. In other words, for the purposes of appraising BPML categories in supporting CN requirements, we should consider the aspects related to both the BPMLs and the CNs simultaneously. Therefore, our designed evaluation methods as well as our evaluation process are discussed in the following sub-sections, respectively.

4.1 Proposed BPMLs Evaluation Method

Traditionally, the aims of formalizing BPs in an enterprise are twofold: (i) creating their common understanding within the enterprise - so that they can be effectively shared, (ii) supporting their computerized execution - so that software services implementing BPs can be developed in the enterprise.

Research on the paradigm of collaborative networks (CN) addresses different aspects related to co-working among heterogeneous, autonomous, and distributed organizations [3], of which one area of research focuses on how to expand the usage of formalizing BPs from a single enterprise to a network of enterprises, e.g. a Virtual Organization (VO). In the CN however, the twofold aims for BP formalization are slightly different than those in a single enterprise: (i) creating common understanding about the BPs that can be provided by any individual organization in the CN, for the purpose of sharing with others, (ii) supporting the creation of integrated and value-added services on top of the services provided by individual organizations. It is therefore vital in the CNs to have formalized definition of BPs performed by member organizations, and thus selecting the most suitable BPML for this purpose.

Several BP modeling goal-settings have been introduced based on different approaches. For instance, a set of five generic software process modeling objectives have been specified in [5] as follow: “to support process improvement”, “to facilitate human understanding and communication”, “having automated guidance in performing process”, “to support process management”, and “to automate execution support”.

Also, for the non-functional BP modeling requirements, [4] has presented a series of objectives (e.g. the support for discovering of dependencies of processes, the support for change management, etc.). These context-aware objectives still hold today. But for our purposes, for supporting more effective BP collaboration in a CN context, a further goal can be added to these criteria: "to support enterprise collaboration" into this context.

Based on the debate in section 2.1, our primary aim is to focus on supporting collaboration through formalized BPs and evaluating BPML categories for this purpose. Therefore, for preparing the guidelines, a goal-based approach is followed (also known as the objective-based approach) as explained in [10] to extract the collaborative intention aspects within the CN context.

Our goal-based approach has systematically focused on a number of qualitative criteria and indicators, related to the set goals. As the evaluation approach, we adopt the Critical Success Factors (CSFs) method, and follow the requirements for achieving established objectives, by running a Critical Success Factors Analysis (CFA), as explained in [23] and partially described in [33] and [31].

CSF is a classical flexible method to maximize goal achievement, through selecting, applying, and monitoring a few certain factors that are vital for the success in achieving the goal. In other words, after the CSF identification, a set of requirement indicators for monitoring them is provided through the CFA.

CNs have their particularities, and the formalized BPMLs should support achieving CNs' goals. To better characterize the particularities of the CNs, and especially VOs and VBEs, we apply the "Reference model for Collaborative Networks (ARCON)" [3].

We have performed a CFA study, of which the results are provided in the next Section, to find out the CN-compliant CSFs and the key requirement indicators for achieving our goal. This study is based on both technical reviews and experts' opinions. Following the CFA study, we then discuss the evaluation of BPML categories versus the identified and recognized CN requirement indicators.

4.2 CN's Critical Success Factors

According to the ARCON model introduced in [3], and our discussion in section 2, VO/VBE need to manipulate formalized BPs. The following aspects indicate the prominent constitutional objective themes in the CN discipline, as originally extracted from their standard definitions, and later on reviewed and validated by experts from the CN communities (see Section 6.2):

- *Goal-orientation* [focusing on goals through business interactions]
- *Infrastructural Commonality* [supporting infrastructure for co-working and coordination toward goals]
- *Handling Heterogeneity* [supporting non-uniformity in different properties, i.e. operational processes]
- *Network Enabling* [supporting collaboration through computer networks]

The four above-mentioned objective themes are the basic objectives of the CN realization. To attain these objectives, defining and aligning a set of CSFs are inevitable. Rooted in our studies of the CNs and the requirements for a BP modeling tool, we have defined four CSFs as the *primary guidelines* in our CFA process, which are also later validated by the CN experts. First, to enable successful collaboration, the BP modeling tool should provide enough "***comprehensibility***" for partners (e.g. the BP Analysts, IT experts, etc.). The "***ease of use***" is the second issue, which is needed to support convenient and pervasive interoperation through the CNs. The next CSF is "***expressiveness for behaviour***", which is a challenging requirement for enactment of BPs in the CNs. Last, for cost-effective achievement of goals in the CNs, "***accessibility***" of BP documents and standards has to be supported. The coverage of CN's objective and the introduced CSFs are illustrated in Table 2. The "S" in the box at the intersection of rows and columns represent the minimal *Support* between our CSF and the CN's constitutional objective elements. For instance, the table shows that the expressiveness for behaviour minimally supports the CN objective of handling heterogeneity.

Table 2. Intersection of CN's objectives and CSFs

CSFs \ CN's Objectives	Goal-Orientation	Infrastructural Commonality	Handling Heterogeneity	Network enabling
Comprehensibility		S	S	
Ease of Use		S	S	
Expressiveness for behaviour			S	S
Accessibility	S			

4.3 BPML's Key Requirement Indicators

The last step of our CFA process, consists of finding a series of generic/key required indicators from the BP modeling context that can then be used to appraise the suitability of the BPML's categories for the CN purposes. These KRIs constitute the *secondary guidelines* in our CFA process. Similar to the CSFs, these five indicators have also been originally extracted from the literature and the standards [14], and then validated and their definitions modified by CN experts:

***Understandability*:** is the ease of interpretation and capture, by which under specified circumstances, the user can interpret an instance, in order to model, analyse and develop the BP formalism [19]

***Expressability*:** explains the capability to represent the process model's attributes e.g.: controls, resources, flow structures, etc. in an unambiguous way [15]

Flexibility**:** is defined as the ease with which in BP modeling the modifications are possible for types and instances, based on an incomplete level of abstraction [18]. So, partial effects of changes do not necessarily imply complete replacement of BP models [30].

Availability**:** comprises the amount and degree to which business process modeling documents, in specific formats and standards, are accessible and adoptable. Thus, being ready-to-use for desired collaboration by the organizations [21].

Enactability**:** is defined as the ambition of acquiring capability to completely execute the BP model, directly and without exploiting any extra tools and/or information [27].

Fig.4 represents the results of our CFA analysis in a map, indicating the interactions among different CN's objectives, the supporting CSFs, and the KRIs and the types of effect between different entities in CNs (which is represented in the figure by the support arrow). In the next section these KRIs as the secondary guidelines of assessment, are addressed and evaluated for each BPML category.

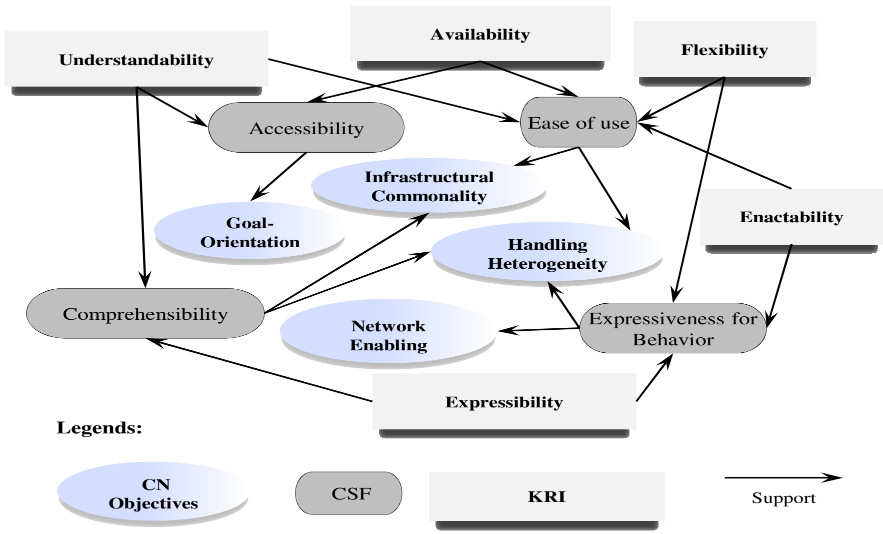


Fig. 4. Interconnections in CFA Diagram

5 Discussion

Our evaluation comprises a two-dimensional descriptive evaluation. The first dimension consists of the BPML categories. Six comprehensive categories are introduced and defined in Section 3.3. The second dimension is the requirements from the CN environment for the BPMLs. Five key requirement indicators are introduced in Section 4.3 for this purpose. Please note that our focus here is on the evaluating BPML categories – instead of individual languages, while the evaluation process involves referencing to individual BPMLs. Nevertheless, in case of suitability found between one BPML category and the CN requirements, there are a number of BPML choices in that category that can be considered for CNs. Due to space constraints, in this section the analysis of six BPML categories versus the five KRIs are only briefly defined and exemplified, pointing only to some of the main distinctive aspects, from the perspective of each requirement indicator.

5.1 Availability

Availability has its roots in reliability notion, which implies the ratio of the time that it takes the users to receive the service to the time specified in prior level agreements [14]. Unavailability of a modeling BP language happens when we do not have “steady-state”, “intervals”, and “user-perceived” availability [21]. For our evaluation, we also assume the availability to represent the existence of BPML documents, which are needed within the context of CNs.

There is an annual research of BPM Market performed since 2005, in [41], which thoroughly surveys the BPM trends. In their published 2012 report, it is stated that the rate of availability for graphical BPMLs is at the highest level, while BPMN is used by more than 60% of all organizations.

Meanwhile, there is less availability for ontological BPMLs (e.g. BPDM). Although, the debate on the timely development of trends is not the focus of this article, but decrease in usage level of BPEL during recent years is noticeable. Even interest and availability of UML and EPC has slightly decreased. Also according to that survey the pervasiveness of the rest of BPML categories (e.g. those in the interoperational and formal categories) are the lowest in usage ranking. So, it is expected that the most fit for organizations that initiate collaboration in CNs would be to apply graphical BPMLs, and especially the BPMN.

5.2 Enactability

Enactability is an important phase in the BPM life cycle. According to [35], after accomplishing “process design” and “system configuration” at the third step of the BPM’s lifecycle “process enactment” is located right before the “diagnosis” step. The more independent is the BPML from the technology and vendor executable environments, the better enactability it has in the CNs, considering its organizations heterogeneity characteristics.

Using the formal semantics supports more effective enactment of the BPs [32], although it slightly reduces the common understandability requirement in CNs. Executional BPMLs enable the enactments of BPs, for sharing BP’s information and automatically executing them through block-based (rule-based) and machine understandable structures [28]. Within the Executional category, BPEL describes behaviour of BPs through representing the interaction between process and its partner, and efficiently supports its orchestration. WS-CDL executional aspects consist of peer-to-peer collaboration of partners from a global point of view, for supporting choreography. But, in this category some languages such as the BPEL have restrictive syntax

[25], which is a limitation for this popular language, and some others, such as YAWL provide exact executional syntax [32].

The formal category of BPMLs - except the embedded notions, e.g. the pi-calculus in WSCDL, provide graphical enactability interface, such as in Reo and Petri net.

Ontological BPMLs, because of their XML supporting structures provide sufficient base for enactability. These languages focus on semantic aspects (e.g. OWL-s), and runs enactment in an abstract level.

Executional issues in interoperational BPML category, where XML enactability is embedded, though properly addressed, still deal with some challenges e.g. the naming and XML reusability in RosettaNet [6], or deficiencies in event handling during interactions [11].

5.3 Expressability

The importance of expressability in CNs arises from the need to concisely express the BPs, so that they can be shared among partners. The expressive power of modeling language represents their ability to express constructs in direct and/or indirect manner [15]. These constructs comprise: control, resources, data, organization, execution, and behaviour of business models. For our purposes in the CNs, expressability encompasses the notion of suitability, which focuses on modeling the BPs, and implies the conformance of a BPML with for instance the 43 workflow patterns, which are introduced in [27]. Although, the evaluated domain in that paper does not focus on the BPML categories, it provides a general inception for comparison of the BPML categories, in our study.

As we map the evaluation of different BPMLs [27] to our proposed categories, we identify that a number of these patterns e.g. the “discrimination”, “milestone”, “partially join”, etc. are relevant kinds of patterns which BP modeling languages and standards have difficulties in expressing them. Nevertheless, we find the languages in the graphical BPML category present better compatibility, while languages in the executional category- except for YAWL- show some deficiencies, for example for supporting “arbitrary cycle”, due to their rigidity in capturing real-world abstractions.

Based on the evaluation in [27], the formal languages category members have good capability for expressiveness, because of their mathematical foundation, e.g. the constructs in Petri-Net; expressive power of workflow pattern design in [35], and the constraint automata used in Reo [42]. Ontological languages use logical basis, for instance in OWL-s, for providing better expressiveness [43].

5.4 Flexibility

Supporting the dynamicity of CNs, the flexibility issues in BPMLs for describing BP’s interaction is vital. BPMLs focus on sustaining their dynamicity in coping with expected and unexpected changes, through adopting flexibility. In [30], four types of flexibility are presented as: *“ design”, ”deviation”, ”underspecification”,* and *“change”.*

For flexibility support, mostly in relation to the first two above-mentioned types, all BPMLs rely on their pre-design notations. Furthermore, the block-based (rule-based) BPMLs could manage the flexibility at higher level (e.g. deviation or underspesification) [18].

In the graphical BPML category, different languages and standards address flexibility differently. In BPMN, flexibility and support for change is provided through the three types of diagram for collaboration, and provision of the concepts of *Pool* and *Lane*. The *Frame* and *Frame Heading* techniques in UML 2.0 Activity Diagram, let the elements of this languages to be defined and described in a modular and flexible structure, and thus flexibility in their “design and deviation” are supported.

Likewise in formal BPML category, mathematical concepts help to retain model’s identity; for instance the structure supporting atomic and complex activities in PSL, as well as the graphical representation of concepts in Petri-Net and Reo facilitates the modification flexibilities, and thus support flexibility in “design and deviation”.

Applying the XML structures usually supports flexibility in design, changes, and even the underspesification to certain extent [30]. YAWL, BEPL (through its interrelations), and WSCDL (through its choreography) support various types of flexibility. Even RosettaNet’s PIP techniques, channelize the modifications.

Languages in the ontological BPML category support flexibility at sufficient level, which support modification through the primary definition of BPs (e.g. process model definition in OWL-s).

5.5 Understandability

Understandability facilitates the BP acquisitions and interactions among CN’s stakeholders. This notion has been reviewed and analysed in different research and especially against the complexity notion, as the other extreme. Generally, understandability comprises the following two aspects, as mentioned in [19]:

- Model-related factors, which affect the understandability of the defined model, e.g. unambiguity, simplicity.
- Person-related factors, which have close relations to the background knowledge and experience of participants.

In our study for the CNs, we mainly address the second aspect above, and focus on human understandability. We think that the first aspect of understandability above is more reflected through our Expressibility KRI.

Although, the understandability of the BPMLs has been reviewed frequently in the past, and a number of guidelines are defined for BP modeling, e.g. the smaller size of the model makes it better for understanding; or the higher degree of input and output defined for one element in a model causes the more complexity for understandability, etc. But, we consider that: the ease of “comprehension of the features introduced in a model”, “ability to present/model concepts without error”, and “labelling less ambiguous” [20] constitute the main understandability’s principles in BPMLs.

With these criteria, generally the graph-based languages are more understandable than the rule-based ones [18]. That is also the reason why they have become more popular at enterprises. However, among the graphical standards, BPMN is more complex for understanding, when compared to the UML and EPC [13].

On the other hand, the understandability of the executional and ontological BPML categories, due to providing less interaction with the human side, is under criticism. Also, the interoperational standards (e.g. the PIP's technical knowledge in Rosettanet) are at a more abstract level for understandability [6]

6 Conclusion

Following the evaluation approach addressed in section 2.3, we have now reached the final step, namely the step 6 in Fig.1, of which the results are provided in this section. We shall again note that our findings are prepared as the consequence of a systematic but subjective method, and our produced results are examined and validated by experts in the field of CN, as explained below.

6.1 Evaluation Results – BPMLs for CN Context

Grounded in our roadmap and the goal-based approach which we followed using the CFA method, we have identified six key requirement criteria for measuring the collaboration-aware adoptability of BPMLs in CNs. The result of our extensive evaluation in previous sections is summarized in Table 3. As shown in the table, BPMLs are ranked through four levels of support that they can provide to the CN, namely: *Strong*, *Sufficient*, *Moderate,* and *Not sufficiently addressed* levels, show the suitability of BPMLs in the CN context.

Table 3. Final Evaluation of BPMLs' suitability in CNs

KRIs for BPMLs Supporting CN / BPML Categories	Understandability	Expressibility	Enactability	Availability	Flexibility
Graphical	++	+-	+-	++	+-
Formal	--	++	+-	--	+-
Executional	+-	+-	++	+-	+-
Ontological	+-	++	+-	+-	+-
Interoperational	--	+-	++	+-	+-
Monitorial	N	N	N	N	N

Legend:
++: Strong support
+-: Sufficient support
--: Moderate support
N: Not sufficiently addressed

Considering our discussions in previous sections, the graphical BPML category provides strong advantages of understandability and availability, and sufficient support for all other criteria.

Due to complexity of their user interaction, the formal languages are not pervasive. But they are strong in expressibility and should be considered as the supporting layer for soundness of BP expression next to the graphical modeling languages.

Executional category is strong in enactability. Also flexibility is one of their considerable features, besides having less ambiguity in modeling real world. Although the lack of interactive graphical depiction, needed for less technical users, is yet a serious criticism.

Ontological languages, due to their well-defined semantics, and focus on graphical and executional aspects, are desirable but not yet sufficiently mature and popular. Interoperational BPML category is just used for supporting collaboration, and mostly emphasizes on interactions instead of abstract BP modeling from real world. Also understandability problems for users cause serious concerns. Monitorial BP Languages are not practically fitting in this context to evaluate for CNs, but promising to be observed in future.

6.2 Results Validation

To evaluate and validate the results of our subjective review of suitability of the BPMLs for collaboration among organizations involved in CNs, we have performed a two-phase survey through questionnaires responded by recognized CN experts. For this purpose, we specifically targeted international experts from two well-known CN communities, namely the PRO-VE[1] (main CN-related conference of 16 years) and the SOCOLNET[2] (Society of Collaborative Networks, founded in 2005). We have approached a number of recognized members in these two communities, to benefit from their past experience and opinion on using BPMLs in the CN context.

The first phase of our survey focused on inquiring about appropriateness of both our introduced critical success factors (CSFs) for BPMLs and the key requirement indicators (KRIs) for evaluating BPMLs in the CN context. We have used the results of this evaluation phase to on one hand to establish and enhance our categorization process of BPMLs and on the other hand identify the main specific KRIs for evaluating BPMLs in the CN context.

The second phase of our survey encompassed evaluation of our final results in relation to both the suitability of our defined BPMLs categories and modeling languages assignment in each category, as well as validation of our final comparison findings and our subjective rating of BPML categories versus the KRIs criteria.

During the two iterations, twenty-eight filled questionnaires were processed from the responders from different countries, including France, Norway, Portugal, Brazil, Italy, Poland, Iran, USA, Greece, Finland, Tanzania, and the Netherlands. The results of our two-phase survey are summarized in Fig.5.

Parts “a” and “b” of Fig.5, indicate the evaluation results of our intermediary long lists of CSFs and KRIs by the experts. The uppermost ranked items in these lists, with more than fifty percent acceptance scores are then selected for conducting our second phase of the survey.

In the second phase, in part “c” the evaluation results of appropriateness of our introduced BPML categories, and the members we have assigned to each category are

[1] `http://pro-ve.org`
[2] `https://sites.google.com/a/uninova.pt/socolnet/`

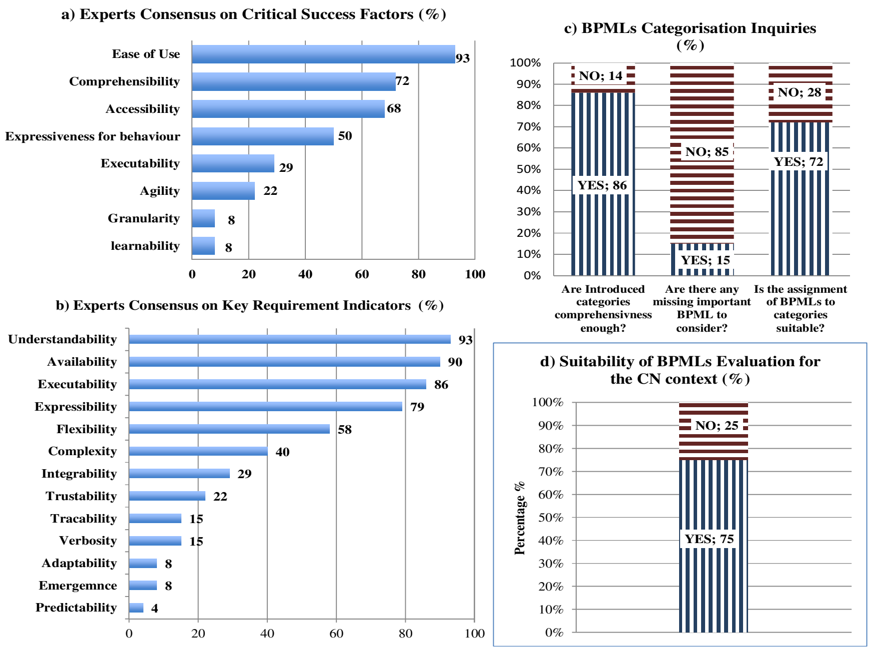


Fig. 5. Validation Summary

rated by the experts. In part "d", final results of our study, and ranking of BPML categories against CN requirements are evaluated by experts. As illustrated, our produced results are evaluated and approved positively by experts, with a more than 72 percent consensus scores. We consider these expert evaluation results as the underpinnings of our evaluation process, and the validation of its outcomes.

6.3 Epilogue

In this article, we presented an analysis and evaluation of the categories of Business Process Modeling Languages in support of Collaborative Networks, namely their suitability for supporting collaboration among enterprises.

Extending the method introduced in [7], we have developed a roadmap for evaluation of BPML categories in the CN context. Following this roadmap, we first "*focus*" on categories defined in previous research for BPMLs. Then we define the "*contextual model*" based on the established CN reference model (ARCON). During our "*review"* step we introduce and provide detailed description for six BPML categories. Thereafter, as the guidelines for evaluation, a set of "*primary guidelines*" is provided through identification of the CSFs, followed by a set of "*complementary guidelines*" that is specified through the introduction of KRIs for CNs. In the last step of the method, the "*context analysis*" is performed, through discussing and analysing the introduced BPML categories versus the KRIs, which are required in the CNs, and summarizing the results.

For validating our findings in this study, at the last stage, a two-phase questionnaire survey is conducted, of which their results appropriately support our conclusions. This survey is performed with the involvement of well-known experts and professionals in Collaborative Networks area.

Due to partially adopting a qualitative analysis approach to the reviewing and evaluating BPMLs, our conclusions are not fully objective. Nevertheless, our results in table3 represent the most appropriate categories of BPMLs that can support each aspect and requirement for collaboration.

For instance, applying the results gained in our evaluation approach, for modeling processes in typical CN contexts, the domain experts may take advantage of understandability and availability of graphical BPMLs, and thus choose the BPMN for practical reasons. However, depending on the environment requirements other categories might be much more suitable. As an example, for the technical BP expressibility purposes for the BP integration, the formal and ontological BPML categories are more suitable than others for adoption in VBEs and VOs. Therefore, the OWL-s or one of the other formal languages might be preferred. A more advanced line of follow up research shall follow on BPML customization, to address ambiguity resolution and semantics enrichment.

Acknowledgment. This work is partially supported by the FP7 project GLONET, funded by European Commission.

References

1. Afsarmanesh, H., Camarinha-Matos, L.M., Msanjila, S.S.: Models, Methodologies, and Tools Supporting Establishment and Management of Second-Generation VBEs. IEEE Transactions on Systems, Man, and Cybernetics 41(5), 692–710 (2011)
2. Afsarmanesh, H., Ermilova, E.: The management of ontologies in the VO breeding environment domain. Int. J. of Services and Operation Management 6(3), 257–292 (2010)
3. Camarinha-Matos, L.M., Afsarmanesh, H.: On Reference Models for Collaborative Networked Organizations. International Journal of Production Research 46(9), 2453–2469 (2008)
4. Chung, L., do Prado Leite, J.C.S.: On non-functional requirements in software engineering. In: Borgida, A.T., Chaudhri, V.K., Giorgini, P., Yu, E.S. (eds.) Conceptual Modeling: Foundations and Applications. LNCS, vol. 5600, pp. 363–379. Springer, Heidelberg (2009)
5. Curtis, B., Kellner, M.I., Over, J.: Process modeling. Communications of the ACM 35(9), 75–90 (1992)
6. Damodaran, S.: B2B integration over the Internet with XML: RosettaNet successes and challenges. In: Proceedings of the 13th International World Wide Web Conference, pp. 188–195 (2004)
7. Ulrich, F.: Evaluating modelling languages: Relevant issues, epistemological challenges and a preliminary research framework (1998)
8. De Nicola, A., Lezoche, M., Missikoff, M.: An Ontological Approach to Business Process Modeling. In: Proceedings of IICAI (2007)
9. Aguilar-Saven, R.S.: Business process modelling: Review and framework. International Journal of Production Economics 90(2), 129–149 (2004)

10. Goldkuhl, G., Lagsten, J.: Different roles of evaluation in information systems research. In: IT Artefact Design and Work Practice Intervention (2012)
11. Green, P., Rosemann, M., Indulska, M., Manning, C.: Candidate Interoperability Standards: An Ontological Overlap Analysis. Data & Knowledge Engineering 62(2), 274–291 (2007)
12. Havey, M.: Essential business process modeling. O'Reilly Media (2009)
13. Indulska, M., zur Muehlen, M., Recker, J.: Measuring Method Complexity: The case of Business Process Modeling Notation, Report 09-03 (2009), `http://www.BPMcenter.org` (viewed on December 2012)
14. ISO, International Organization for Standardization, ISO/IEC/IEEE24765, Systems and Software Engineering Vocabulary (2010E)
15. Kiepuszewski, B., ter Hofstede, A.H.M., van der Aalst, W.M.P.: Fundamentals of Control Flow in Workflows (Revised version), QUT Technical report, FIT-TR-2002-03 (2002)
16. Ko, R.K., Lee, S.S., Lee, E.W.: Business Process Management (BPM) Standards: A Survey. Business Process Management Journal 15(5), 744–791 (2009)
17. Zdun, U., Hentrich, C., Van Der Aalst, W.M.: A Survey of Patterns for Service-Oriented Architectures. International Journal of Internet Protocol Technology 1(3), 132–143 (2006)
18. Lu, R., Sadiq, W.: A Survey of Comparative Business Process Modeling Approaches. In: Abramowicz, W. (ed.) BIS 2007. LNCS, vol. 4439, pp. 82–94. Springer, Heidelberg (2007)
19. Mendling, J., Reijers, H.A., Cardoso, J.: What makes process models understandable? In: Alonso, G., Dadam, P., Rosemann, M. (eds.) BPM 2007. LNCS, vol. 4714, pp. 48–63. Springer, Heidelberg (2007)
20. Mendling, J., Reijers, H.A., van der Aalst, W.M.: Seven Process Modeling Guidelines (7PMG). Information and Software Technology 52(2), 127–136 (2010)
21. Milanivic, N., Milic, B., Malek, M.: Modeling Business Process Availability. In: Proc. IEEE Congress on Services (SERVICES), pp. 315–321 (2008)
22. Mili, H., Tremblay, G., Jaoude, G.B., Lefebvre, E., Elabed, L., Boussaidi, G.E.: Business Process Modeling Languages: Sorting Through The Alphabet Soup. ACM Computing Surveys (CSUR) 43(1), 4 (2010)
23. OASIS: Reference Architecture for service Oriented Architecture version 1.0 (2008), `http://docs.oasis-open.org/soa-rm/soa-ra/v1.0/soa-ra-pr-01.pdf.`
24. Papazoglou, M.P., van den Heuvel, W.J.: Service-Oriented Design and Development Methodology. International Journal of Web Engineering and Technology 2(4) (2006)
25. Recker, J., Mendling, J.: On The Translation between BPMN and BPEL: Conceptual Mismatch between Process Modeling Languages. In: Proceedings of the 18th International Conference of Advanced Information Systems Engineering, pp. 521–532 (2006)
26. Roser, S., Bauer, B.: A Categorization of Collaborative Business Process Modeling Techniques. In: Seventh IEEE International Conference on E-Commerce Technology Workshops, pp. 43–51 (2005)
27. Russell, N., ter Hofstede, A.H.M., van der Aalst, W.M.P., Mulyar, N.: Workflow Control-flow Patterns: A Revised View, BPM center report, BPM-06-29 (2006), `http://www.BPM-Center.org` (viewed in December 2012)
28. Russell, N.C.: Foundations of Process-Aware Information Systems, Ph.D. Dissertation, Queensland University of Technology (2007)
29. Soleimani Malekan, H., Afsarmanesh, H.: Business Process Modeling Languages Supporting Collaborative Networks. Proceeding of ICEIS 3, 258–266 (2013)

30. Schonenberg, H., Mans, R., Russell, N., Mulyar, N., van der Aalst, W.: Process flexibility: A survey of contemporary approaches. In: Dietz, J.L.G., Albani, A., Barjis, J. (eds.) CIAO! 2008 and EOMAS 2008. LNBIP, vol. 10, pp. 16–30. Springer, Heidelberg (2008)
31. Sudhakar, G.P.: A model of critical success factors for software projects. Journal of Enterprise Information Management 25(6), 537–558 (2012)
32. Ter Hofstede, A.H.M., van der Aalst, W.M.P., Adams, M., Russell, N.: Modern Business Process Automation: YAWL and Its Support Environment. Springer (2010)
33. Trkman, P.: The Critical Success Factors of Business Process Management. International Journal of Information Management 30(2), 125–134 (2010)
34. Mendling, J., Neumann, G., Nüttgens, M.: A comparison of XML interchange formats for business process modelling. In: Workflow Handbook, pp. 185–198 (2005)
35. van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M.: Business process Management: A survey. In: van der Aalst, W.M.P., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg (2003)
36. Vergidis, K., Tiwari, A., Majeed, B.: Business process analysis and optimization: Beyond reengineering. IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews 38(1), 69–82 (2008)
37. Van der Aalst, W.M., Dustdar, S.: Process Mining Put into Context. IEEE Internet Computing 16(1), 82–86 (2012)
38. Lee, J., Siau, K., Hong, S.: Enterprise Integration with ERP and EAI. Communications of the ACM 46(2), 54–60 (2003)
39. Weske, M.: Business Process Management: Concepts, Languages, and Architectures, pp. 5–25. Springer, Leipzig (2007)
40. Melão, N.: E-business processes and e-Business Process Modelling: A state-of-the-art overview. International Journal of Services Technology and Management 11(3), 293–322 (2009)
41. Wolf, C., Harmon, P.: State of Business Process Management (December 2012), `http://www.bptrends.com`
42. Arbab, F.: Reo a channel-based coordination model for component composition. Math Structure in Computer Science 14, 32–366 (2004)
43. W3C, OWL-S: Semantic Markup for Web Services (2004), `http://www.w3.org/Submission/OWL-S` (April 2012)
44. Devlin, B.A., Murphy, P.T.: An architecture for a business and information system. IBM Systems Journal 27(1), 60–80 (1988)
45. Georgakopoulos, D., Hornick, M., Sheth, A.: An Overview of Workflow Management: From Process Modeling to Workflow Automation Infrastructure. Distributed and Parallel Databases 3(2), 119–153 (1995)

Creating a Business Case from a Business Model

Lucas Onno Meertens, Eelco Starreveld, Maria-Eugenia Iacob, and Bart Nieuwenhuis

Department of Industrial Engineering and Business Information Systems, University of Twente, Enschede, The Netherlands
lucas@lmeertens.nl, eelcostarreveld@gmail.com, {m.e.iacob,l.j.m.nieuwenhuis}@utwente.nl

Abstract. Intuitively, business cases and business models are closely connected. However, a thorough literature review revealed no research on the combination of them. Besides that, little is written on the evaluation of business models at all. This makes it difficult to compare different business model alternatives and choose the best one. In this article, we develop a business case method to objectively compare business models. It is an eight-step method, starting with business drivers and ending with an implementation plan. We demonstrate the method with a case study for innovations at housing associations. The designed business case method can be used to compare and select the best business model successfully. In doing so, the business case method increases the quality of the decision making process when choosing from possible business models.

Keywords: Business Case, Business Model, Method.

1 Introduction

Due to shortening product lives, intense global competition, a disruptive and agile [1]. In addition, the chosen course of action is of great importance for the future performance of organizations. With the renewal of business models, multiple possible directions can be defined. A recent example is seen in the automotive industry. Car manufactures need to choose if they want to produce cars running on alternative energy, and next, which type of energy. Hybrid, bio-fuel, electric, or hydrogen are all options. Making the choice is hard, for each of the alternatives require a business model change and the success of the produced car is unsure. This is an example of the need for a method to objectively compare alternative business models, and choose the best course of action.

A business case can be of help to form the answer to this question. A business case is a tool for identifying and comparing multiple alternatives for pursuing an opportunity and then proposing the one course of action that will create the most value [2]. Making a business case for the possible business model alternatives, gives the decision makers a solid and objective as possible basis, to make the best choice [3].

Choosing one of the business model alternatives, should be well considered. Instead of a gut feeling, each of the alternative's consequences, impact, risks, and

B. Shishkov (Ed.): BMSD 2013, LNBIP 173, pp. 46–63, 2014.

benefits for the organization, should be assessed as objectively as possible. This will result in a better choice, and better organizational performance.

However, the main problem is that it is unclear how alternative business models can be compared to choose the best course of action. A business case could be one of the solutions, for it compares alternatives in terms of costs, benefits and risks. Existing problems are that it is unclear how a business case should be made from a business model. Also, it is unclear what good business case components are, and which business model components are of relevance for the development of the business case.

2 Methodology

The research design is based on the design science research methodology (DSRM) [4]. This method is chosen because it creates an artefact as solution to a problem. In this research, the problem is the unstructured decision making of potential business models. The artefact designed is a business case method which enables objective comparison of business models. Further, the DSRM enables process iterations, so that it is possible to adjust previous phases to increase the quality of the artefact. However, because the review of academic literature is less emphasized, the method is adjusted to include the valuable academic literature in the process. For the literature study, the five-stage grounded theory method for rigorously reviewing literature by Wolfswinkel et al.[5] is used. This method assures solidly legitimized, in-depth analyses of empirical facts and related insights, including the emergence of new themes, issues and opportunities[5]. **Fig. 1** shows the five sequential steps integrated with the DSRM method.

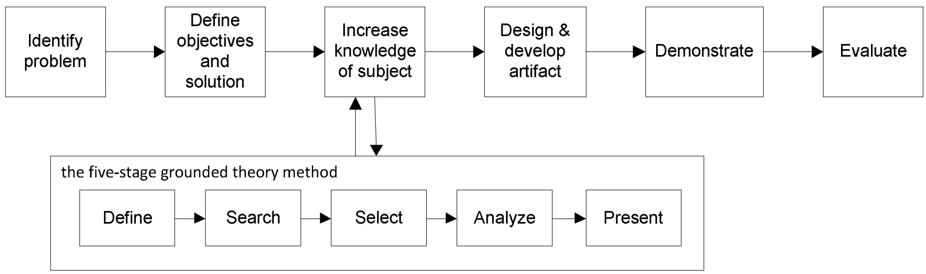


Fig. 1. DSRM process of Peffers et al. [4] with the grounded theory method from Wolfswinkel et al. [5]

Starting with the first step of the DSRM [4], the introduction to this article identifies the problem. Namely, the need to objectively compare business models. Following the DSRM, we identify the research objective: design a structural method to create a business case of business models, to be able to objectively compare the assessed business models, and choose the best alternative. We present the literature review of business cases and business models, which increases our knowledge on the subject, elsewhere.

3 The Business Case Method

This section creates a new artefact in the form of a business case method. The design of our business case method is based on the two approaches identified by literature review [2], [6]. Both of them have a list of components. These lists partly overlap, yet each has distinct advantages and disadvantages. Based on the comparison of these two approaches, eight main components can be identified, which Table 1 lists.

Table 1. Components of the business case method

1. Business driver	The cause, problem, or opportunity that needs to be addressed
2. Business objectives	The goal of the business case stating which objectives are aimed for
3. Alternatives	Representing the options to reach the objectives
4. Effects	Positive and negative effects that come with the pursued alternative
5. Risks	Risks that come with the pursued alternative
6. Costs	Costs that come with the pursued alternative
7. Alternative selection	Based on gathered data the best alternative is chosen
8. Implementation plan	Plan which explains when and how the alternative is implemented

In contrast to the business case method proposed by Ward et al. [6], this method does take alternatives into account, similar to the model of Harvard Business Review Press [2]. This is because in most cases more than one solution can be thought off and applied to reach the goal. Therefore, it would be bad to go with the first possible solution without putting some effort in the quest for other compelling solutions.

Furthermore, the fourth point, alternatives, is different from the business case methods proposed in the reviewed literature. There, the authors only look to the benefits that the proposal brings. Of course, the benefits are important for the business case. The possible negative effects, however, cannot be dismissed. Therefore, a good overview of not only the benefits but also the disadvantages should be presented in the business case as an overview of the caused effects of the proposed project. According to Ward et al. [6], organizations who overstate the benefits to obtain funding are the least likely to review the outcome and less than 50% of their business case projects deliver the expected benefits resulting in unsatisfied senior management.

As the components are the main concepts of the proposed method, we clarify all eight of the components individually in this section.

3.1 Business Drivers

The meaning of the business drivers originates from the business case method by Ward et al. [6] and has not changed. The business drivers stand for a statement of the current issues facing the organization that need to be addressed. These can either be problems or opportunities and ideas with enough potential to make it worth pursuing. Applied to business models, the business driver is most likely to originate from the need for business model innovation. Chesbrough [1] argues that due to shortening product lives, even great technologies can be relied upon no longer to earn a satisfactory profit before they become commoditized. Practice has learned that even great business models do not last forever. Therefore, he argues, a company needs to think hard about how to sustain and innovate its business model. For future markets will be smaller, more highly targeted (and effective), and the new environment will require different processes to develop and launch products successfully.

3.2 Business Objectives

The business objectives are the goals of the innovation. Both methods discussed in the theoretical framework advice to set business objectives. They state which business drivers are addressed and how these are hoped to be achieved with the proposed project. This can be one or more specific aspects of the strategy that need to be improved or modified; one or more of the business model components that need improvement; or processes or products that need to become more efficient and better address the needs of customers.

3.3 Alternatives

The alternatives represent the available options to reach the objectives. At the start of this section, we describe the reasoning to include identification and assessment of alternative solutions in the method. Summarized, the argument is that it would be unwise to go with the first idea that comes along that addresses the business drivers, without investigating whether other, perhaps better, alternatives exist.

Sometimes, the benefits of a single specific opportunity or idea are assessed. In such cases, it might be hard to find a substitute or alternative to the opportunity. Thinking of alternatives and assessing them increases the chance of pursuing a better-balanced alternative, instead of the first that comes to mind. All alternatives need to be compared with the current situation.

Amongst others, identification of alternatives can be done by assigning a senior manager with the task to define and launch business model experiments [1]. Harvard Business Review Press [2] proposes brainstorm sessions as a tool to identify alternatives. Both tools can be used to identify alternative business models. Next to those tools, market assessment tools or SWOT analysis may be suitable to come up with alternatives.

3.4 Effects

The effect component is the largest of all. This is because a variety of actions needs to be performed with the effects to create a consistent and structured overview of the effects on the organization per alternative. Effects are the positive (benefits) and negative (disadvantages) effects that an alternative causes. First, effects need to be identified. Second, it is important to come up with measures for each effect. Third, each effect must be connected to an owner. This increases involvement with the project within the organization, and stimulates owners of benefits to help establishing the alternative when it is approved. Fourth, each effect needs to be placed in the framework in Table 3 [6]. For each effect, the framework determines the type of organizational change (do new things, do things better, or stop doing things) and the degree of value explicitness (from observable to financial).Fifth and final, a time frame is estimated per alternative. This time frame gives information of when the project starts, when it delivers results, and when the project finishes. Each alternative goes through these five steps.

3.5 Risks

The fifth component is concerned with risk assessment of each alternative. Risk is defined as the probability that input variables and outcome results vary from the originally estimate [7]. How risks are assessed depends on the situation and needs further research per case. Amongst many others, the “best case/worst case scenario” method can be used to assess the risk of the alternatives. With this method, two scenarios are developed and the effects of each scenario on the organization are estimated. In the first scenario, the alternative will perfectly result in the expected benefits. In the second scenario, the worst reasonable possible situation will evolve caused by the alternative.

3.6 Costs

Costs are one of the most important aspects of a business case. The costs give an indication of the total expected investment costs, and expected profit over a specific time period. The investment costs represent the money needed to implement the business model change in the organization. Also, in the costs section, the expected payback time is calculated to indicate how long it will take for the break-even point is reached.

3.7 Alternative Selection

After gathering the data for all alternatives in the previous steps, the best option can be chosen by weighting the expected effects against the expected calculated costs. Harvard Business Review Press [2] suggests that the best alternative is partly chosen based on feelings. However, if the risks are translated into expected costs, this can be added to the costs-effect equation. Then the alternatives have to be compared based on the non-financial effects and the total expected costs/profit of the alternative. Many methods to do this exist, varying from complex to rather simple. For example, the direct-rating method, point-allocation method, and analytical hierarchy process [8].

A rather simple three-step method could be derived from the direct-ranking method. First, all effects and cost/profit numbers are listed together. Second, positive effects and profit are ranked according to importance relative to each other from “0-100”. Negative effects and costs are ranked relative to each other on a scale from

"-100 – 0" as well. Third, the values of the effects and cost/profit per alternative summed. The alternative with the highest total score wins.

3.8 Implementation Plan

Now that the best alternative is selected, it is important to develop a plan of action. Tasks, roles, objectives, resources, dates, and responsibilities are parts of this implementation plan. The level of detail of an implementation plan varies depending on the case. The plan lays out how progress can be tracked and success measured when the proposed solution is put into action. Without this, actual success of a business case is hard to verify.

4 Connecting the Business Case Method to Business Modelling

In this section, the developed business case method is applied to the business model concept. **Fig. 2** visualizes the connection. The figure shows the business case steps on the left. The sources, types of information, or input for each of those steps are on the right.

The first step contains the business driver. Business drivers for business model innovation can come from different sources. In general, shortening product lives, intense global competition, and the disruptive and agile environment are the main sources for business drivers [1]. This can lead to one of the three causes for business model renewal. The business objective represents the goals that the business model change aims to obtain.

The next step is identification of alternatives. In this step, multiple business models can be developed with the focus on meeting the business objectives. Next, the effects, risks, and costs of each of the business model alternatives are assessed. The effects represent the positive and negative non-financial effects that alternatives cause. The effects can be represented with a framework for business case development [6].To assess the risks of the project, one of the risk assessment methods described in literature for project management can be used. The risk assessment part should at least cover the points of Remenyi [7]. The risk can be represented in a probability vs. impact matrix.

Often, the expected financial benefits, along with the costs of the project, are the most important part for decision makers using business cases. In the costs section, changes in the business models costs and revenue component need to be assessed. The cost component of a business model must cover costs created in other components, such as key activities. Next to the expected costs and profits, the payback period and return on investment should be presented.

Using a multi-criteria method, the most suitable business model can be selected in the seventh step. After that, an implementation plan can be developed. During step three till eight, alternative business models should be compared to the current business model to assess the changes and effects that it causes. For example, in the fourth step, only the effects that differ from the current business model are assessed.

The reason for this is that the other effects remain the same for both alternatives, and thus only increases the size and complexity of the business case.

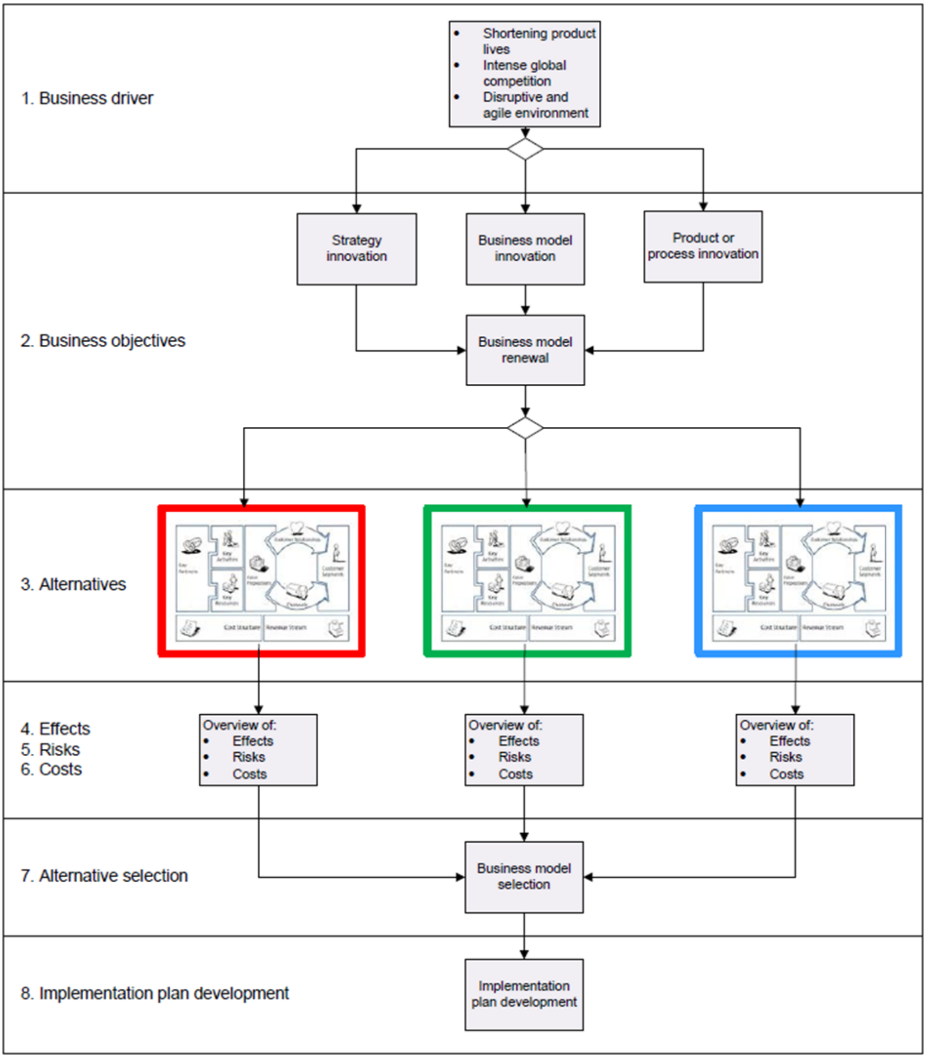


Fig. 2. Business modelling connected to the business case method

5 Method Demonstration and Evaluation: DEA Logic and Housing Associations

Having created the artefact (business case method), the next step is to demonstrate it. We use a case study of the company DEA Logic, which provides products and services for Dutch housing associations. The main two stakeholders in the case are the company DEA Logic and the Dutch housing associations. The innovation is developed by DEA Logic, and the target customers for this innovation are Dutch housing associations. The innovation will have an impact on the business model of the Dutch housing associations.

DEA Logic is an engineering company specialized in advanced electronics, security software, and consulting in information technology, information management, and building management. Over the last years, DEA Logic developed an access control system called C-Lock, which has a major position in their product portfolio currently. The C-Lock system can be extended with multiple solutions. This way, apartments can be better adjusted to the needs of the tenants. In this case, DEA Logic wants to discover whether their product is favourable for (Dutch) housing associations. A business case needs to be developed.

In the Netherlands, a housing association is a non-profit organization, which' mission is to build, manage, maintain, and rent houses and apartments. The responsibilities are defined and assigned by the Ministry of the Interior and Kingdom Relations. Each housing association is private, but can only operate within boundaries set by the Dutch government. Therefore, housing associations do not differ much. In addition, all housing associations have more demand than supply currently, which causes waiting lists. The houses they rent are favourable for citizens with a low income (an annual income of € 43.000 is the maximum). The associations are tasked to supply good housing possibilities for the relatively more vulnerable and poorer people in society. Similar constructions exist in other countries. For example, the United Kingdom has government-regulated housing associations with the same goal; to provide housing to people on a low income or people who need extra support.

Thanks to the public character of the housing associations, all needed information for this case is public and presented on websites of housing associations, the government, and the central fund for people housing. For the scope and purpose of this research, applying the DEA Logic case on Dutch housing associations in general is sufficient to demonstrate the designed method.

The data and numbers used in the business case are based on calculations by DEA Logic, and internet sources. For reasons of confidentiality, the numbers are not accurate. The business case gives an indication of the order of magnitude of the costs difference between the two discussed alternatives. If in the future, a housing association would like to realize the project, a new business case has to be made, to assess the effects of the innovation on their specific situation. For the purpose of demonstrating the business case method, the used numbers and accounted variables are sufficient.

DEA Logic develops technological and electronic innovations for real estate amongst others. The C Lock access control system is one of those products. The latest innovation for newly built or renovated apartment buildings is IP-infrastructure. In the current situation, each apartment in a building complex is supplied with public utilities and digital infrastructural connections. In the Netherlands, each apartment is provided with at least a telephone line, television cable, intercom system, and often fiberglass connection for internet. Each of these connections makes use of their own wires. The main idea of IP-Infrastructure is to supply each apartment with only one

TCP-IP connection, combining telephone, television, intercom, and internet, as well as other possible data connections.

This infrastructure not only reduces infrastructural costs and materials of newly built or renovated apartments, but also increases the amount of possible functionalities. The currently developed functionalities are derived from the C-Lock access system, and can be connected to the receiver easily. Tenants can choose individually which solutions they need. The core of the innovation is to increase apartments' flexibility, functionality, and luxury, and to minimize the maintenance costs. Table 2 gives examples that help with this.

Table 2. Examples of IP-infrastructure functionality

Access	Electronic keys are used to grant access to the building and the apartment. If favourable, the system can be extended with an automatic door opener, which opens the door if the tenant with the right key is standing in front of the door.
Intercom	This is a door phone system with video support as seen in most newly build apartment buildings nowadays.
Security	This module contains a burglar alarm, smoke detector, and camera monitoring. In case other tenants also have this module, the alarm message can also be send to them, for example in case of a fire.
Care	The intercom phone with touchscreen system can be extended with additional modules for extra functionalities. This could include personal alarm, telemedicine, telemonitoring, and even location detection to prevent people from wandering off.
Communication	Currently, tenants and housing associations communicate by letters or phone. With the communication solution, housing associations can send information through the intercom system, for example about maintenance. Tenants are also able to send requests for maintenance using the intercom system.

The C-Lock and IP-Infrastructure innovations by DEA Logic are suitable for Dutch housing associations, for they build, rent, manage, and maintain apartments for a diverse target group. The target group is diverse, as their customers are young as well as old people. In addition, families with children and people who need daily nursing support belong to the target customers. Introducing DEA Logic's innovations increases the suitable target group for each apartment, as it can be adjusted to the needs of the tenant more easily. Furthermore, the use of IP-infrastructure decreases maintenance costs.

The innovations affect the housing association's business model. Renting out C-Lock solutions and IP-infrastructure becomes a new key activity. DEA Logic becomes a new key partner, together with several service providers. Also the value proposition is extended, for apartments are more secure and luxury. The suitable customer segment for each apartment increases, as it can be adjusted to the needs of various tenants. Finally, a new revenue stream is added, for the IP-infrastructure is rented out, in combinations with C-Lock solutions, in addition to the traditional rent of apartments. Therefore, DEA Logic's innovation and Dutch housing associations form a good combination to test the business case development method.

The following eight paragraphs represent the eight steps of the business case method. We compare two scenarios. In both scenarios, the same apartment complex is built with one hundred apartments. The first scenario represents the current situation. In the second scenario, the IP-infrastructure is implemented together with C-Lock solutions.

5.1 Business Drivers

Based on the vision and strategy of the three largest housing corporations [9], their mission is to build, manage, and maintain quality tenement housing for people with a low income and vulnerable groups in society. Therefore, it is preferable that building, managing, and maintenance costs of the houses are low. Housing corporations continuously seek possibilities to reduce costs and still deliver high quality, affordable, and luxury homes for a large and diverse target group. IP-infrastructure, in combination with the variety of possible C-Lock solutions provided by DEA Logic, is an innovation that contributes to the corporations' mission.

5.2 Business Objectives

In accordance with the business drivers, the pursued objectives of the IP-infrastructure presented in this business case are the following:

- Reduce maintenance costs
- Increase compatibility with target tenant group
- Increase quality of living environment
- Increase security of tenants
- Increase luxury

5.3 Alternatives

The yellow post-its in **Fig. 3** show the current business model of a Dutch housing association. The value proposition offers low-priced rental houses in a good living environment for people with low income belonging to vulnerable groups in society. Revenue is generated via monthly rent and subsidy from the government.

The blue post-its in **Fig. 3** are additions that show an alternative business model of a housing association with an apartment complex with IP infrastructure. In addition to the current key activities, renting out infrastructure and solutions form a new key activity. DEA Logic becomes a new key partner of the housing corporation, as they provide the solutions and maintain the system. Furthermore, the customer segments are extended with an increased target group including tenants who require special care. The fourth change is in the revenue stream building block. Next to the rent of houses and state subsidy, the housing corporations receive rent for the use of the IP-infrastructure by tenants.

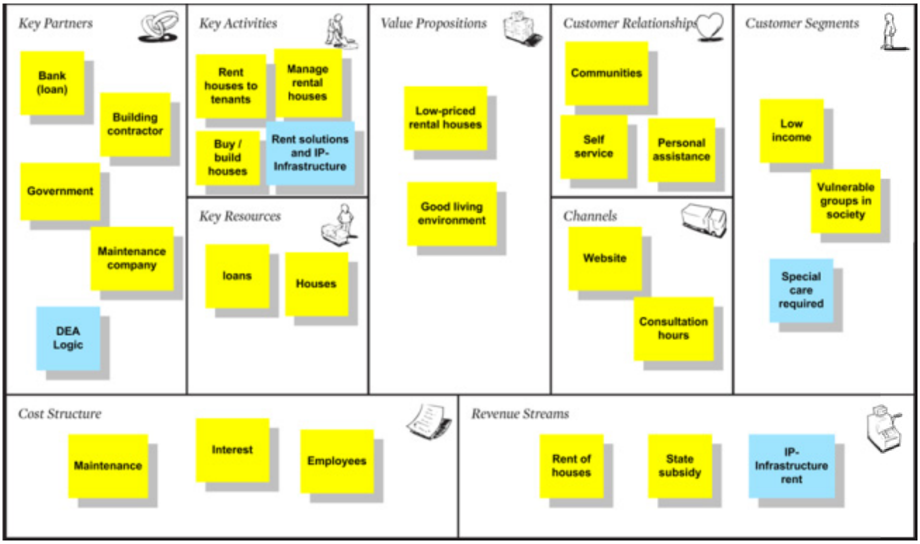


Fig. 3. Business model of Dutch housing associations with IP-infrastructure and C-Lock solutions

Next to changes visible in the business model, many benefits of IP-infrastructure are within the tactical set of the current business model [10]. Therefore, they do not influence or change the business model. However, the resulting business case includes those effects as well.

5.4 Effects

Implementing IP-infrastructure in renovated or newly build apartment buildings affects the organization. Table 3 presents the effects of the new IP-Infrastructure compared to the current, classic infrastructure. The table structures them according to two factors. Horizontally, they are categorized according to the type of required organizational change. Vertically, they are categorized according to the degree of explicitness. Because the only difference between the two alternatives, in terms of business model, is the revenue model, other effects of both alternatives are equal. Therefore, they are represented in only one effects overview table.

Starting with the financial effects, the revenue model behind the IP-Infrastructure is new in the business model. For some solutions, an additional rent is incurred for the use of the IP-infrastructure. Furthermore, the tenants may rent some non-standard C-Lock solutions from the housing association.

The second financial effect is a reduction on maintenance costs. Normally, if an apartment is rented to a new tenant, the door lock and keys are renewed together with the nameplates. With the IP infrastructure, this can be done remotely, saving both time and money. At the main entrance, the touchscreen shows nameplates digitally. Names can be edited from behind the desk by logging into the building's central server. Access rules for the keys can be changed in the same way. New tenants receive new keys with a different RFID chip. Access to the apartment is then only granted by using the new key.

Table 3. Effects of IP infrastructure

Degree of explicitness	Do new things	Do things better	Stop doing things
Financial	Rent C-Lock solutions and IP-infrastructure		Reduce maintenance costs by not replacing door locks & nameplates
Quantifiable			
Measurable		Increased target group Increased security	
Observable	Dependable on non-standardized technology In line with mission and vision	Increase quality living environment	

Two measurable benefits make the organization better. The first benefit is the increase in target groups for apartments. With C-Lock solutions, apartments can be adjusted easily to meet requirements and demands of tenants. For example, if elderly people, who require extra care functions, rent the apartment, a selection of care solutions can be connected to the system, providing the required services. No longer is a specific group of apartments suitable for only a single special target group, but all apartments with the system can be adjusted to be suitable for each target group.

The second benefit is increased security with the IP-infrastructure in combination with C-Lock solutions. Electronic keys are much harder to forge compared to classic keys, keeping unwanted visitors out. Furthermore, with the fire alarm, neighbors are notified as well to be careful and investigate the emergency.

The influences of two observable effects are hard to estimate. First of all, the IP-infrastructure and C Lock solutions are developed by DEA Logic. At the moment, no direct interchangeable alternatives to the DEA Logic's product exist. This makes the apartment building technologically dependent on DEA Logic.

The second effect is the increased quality of the direct living environment for tenants. Each apartment can be fitted with various C-Lock solutions to make living more comfortable. For example, automatically opening doors, curtains, and lights may provide more comfort.

5.5 Risks

As with each innovation, risks are involved. To assess the risks, we use a construction project risk assessment method [11]. This method is suitable, as renovating or building the apartment complex is a construction project. Most risks can be prevented, resulting in a very low overall project risk. However, some risks of the IP-infrastructure alternative remain, due to the following two points:

1. The technology is new. So far, it has been deployed in one apartment building only.
2. The technology is developed and built by one company. The current market does not provide any substitutes that work with the same infrastructure.

These two points are interconnected. A small change exists that the technology does not work as good as was hoped for, or the subcontractor stops supporting the technology. In that scenario, the costs to transform the infrastructure back to the current standard are high. Other risks for both alternatives can either be prevented, or do not have a negative influence on the organization. The total risk of IP-Infrastructure, before prevention, is one and a half times the risk of the classic approach. This is mostly because the classic infrastructure is used almost everywhere and has been improved over time.

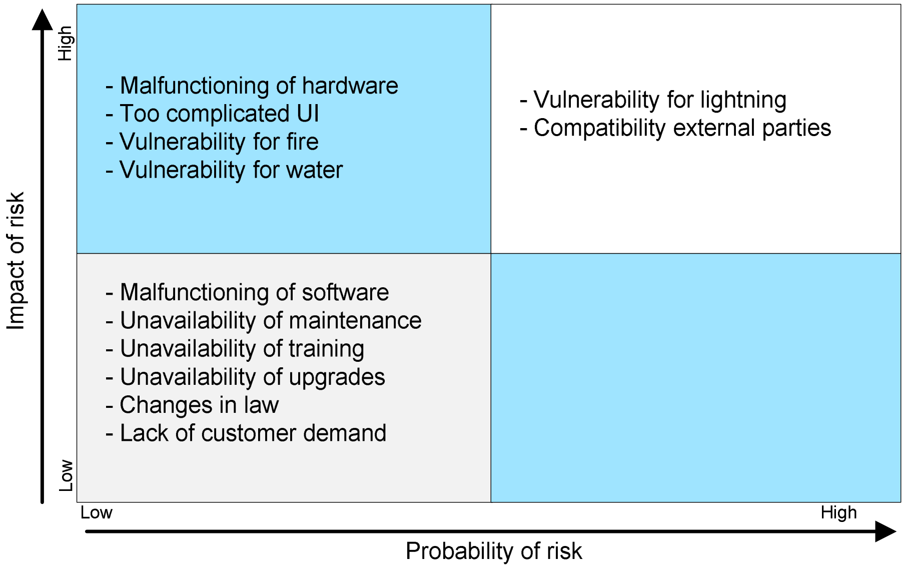


Fig. 4. Risk assessment matrix for IP infrastructure

5.6 Costs

The cost difference, between the current situation and the IP-Infrastructure alternative, depends on two variables. First, the number and type of C-Lock solutions affect the costs. The second variable is time. Time is important, as the housing association's objective is not only to build apartment complexes, but also to maintain them. Therefore, the cost overview also includes maintenance.

To compare the costs of both approaches, an indication of the costs for an apartment complex with 100 apartments is calculated. Only the costs for the infrastructure and the C-Lock solutions are covered. The other building costs are equal for both

Table 4. Estimated costs of construction and maintenance, and estimated profit

Function Costs (€)	Infrastructure		Access		Intercom		Care		Communication	
	Old	New	Old	New	Old	New	Old	New	Old	New
Construction (Initial)	13,000	26,000	30,000	30,000	52,000	50,000	800	400	-	-
Maintenance (Yearly)	500	1,000	11,250	6,950	16,500	7,000	3,600	1,800	750	0
Profit (Yearly)	-	-	-	-	-	-	-	300	-	-

alternatives. Because the costs for construction and maintenance of the infrastructure and the C-Lock solutions vary from situation to situation, several assumptions and raw cost estimates are used.

Table 4 shows estimates of construction costs, yearly maintenance costs, and yearly profit, per function. Next, the maintenance costs and profits are extrapolated over five years to get more insight in the breakeven point of the alternatives. Because of the raw input data, assumptions, and extrapolation of five years, the outcome of this analysis is relatively unreliable and can only be used as an indication for expected costs of both alternatives over a time span of ten years. If the project is deployed in a real situation, more data gathering is needed to calculate the specific values and come to more accurate estimates.

Based on the numbers shown in Table 4, the expected cumulative costs over five years are calculated for both situations. The costs are influenced by both time and functions. More functions leads to more costs, and due to the maintenance costs, over time the total costs increases. The initial costs for the IP-Infrastructure are higher compared to the current situation. However, the difference is not very big, and within three years, the IP-Infrastructure in combination with the access C-Lock solution is cheaper than the current alternative.

In the cost overview, financial differences between IP-infrastructure and the classical approach are assessed. The initial costs for IP-Infrastructure are higher, but due to lower maintenance costs, this difference is equalized within one to three years, depending on the functions. Especially with real estate, long term thinking is important as buildings last for decades. In the cost overview, cost estimates are used. Therefore, they are only extrapolated over five years. However, in case a project is realized with the IP-Infrastructure and building plans are better established and concrete, the costs have to be reassessed to improve reliability before they can be used to make the definitive decision.

5.7 Alternative Selection

The effects, risks, and costs of IP-infrastructure, compared to the classic infrastructure, are discussed in the previous sections. Based on this information, one of the alternatives needs to be selected. Looking at the effects, IP-infrastructure is the best choice as it increases the amount of target groups, quality of living, and security of tenants. Additionally, with the new technology, apartments become more luxury.

The risks, however, are one and half times higher than with classic infrastructure. Then again, this can be reduced strongly using available risk prevention options.

Initial costs of IP-infrastructure are higher, but within four years it becomes cheaper than the classic alternative. Depending on the functions, the estimated IP-infrastructure savings are around € 70.000 after five years. Initial costs are higher, yet maintenance costs are much lower.

IP-infrastructure offers new functionalities and increases security of tenants, quality of living, and target group. Risks are higher, but can be prevented. Initial costs are higher, but money is saved due to the low maintenance costs over time. Therefore, IP-infrastructure is the best alternative to choose.

5.8 Implementation Plan

After their board of directors approves this project, the housing association can implement the project. In this phase, however, it is too far stretched to determine an explicit implementation plan. However, the following can be used as an indication of the steps that need to be made to achieve a successful implementation. The steps are based on the Deming cycle, which is an iterative management method for the control and continuous improvement of processes and products [12].

The first project implementation step concerns planning the project. After the decision to build a new apartment building, the exact installation costs and system specifications can be determined and contracts can be drawn up. In the second step, the apartment building needs to be realized and the IP-infrastructure in combination with C-Lock solutions need to be installed. In the third phase, the system check need to be performed to determine whether the system is secure and works as planned. In the fourth phase, the apartments are rented out to tenants and the solutions can be rented. In addition, problems, flaws, and obscurities need to be analysed.

After this fourth step, the cycle starts again with planning how the determined flaws and problems of the previous phase can be assessed and solved, followed by taking action, checking solutions and implementing them, and assessing if the problems are solved and if others have occurred. If needed, the cycle can start again, until the system is optimized.

6 Discussion

The goal for designing the business case development method to compare business models was to design a method to create a business case of business models, to objectively compare the assessed business models, and choose the best alternative. Because of the abstract descriptive nature of business models, it is often required to involve more tactical and operational details, only implicated by changes in the business model. Deciding which details are useful and which are not must be judged by the maker of the business case. This allows for a certain amount of subjectivity. Table 5 represents which method steps are objective and which are open for subjectivity.

Table 5. Assessment of the objectivity of the business case method

Method step	Objective / Subjective
Business driver	Objective
Business objectives	Objective
Identification of alternatives	Subjective
Effects	Subjective
Risks	Subjective
Costs	Objective
Alternative selection	Objective / Subjective
Implementation plan	Subjective

Business drivers and objectives are fixed input variables. On the other hand, identification of alternatives is variable in most cases. This gives freedom for interpretation and creativity, therefore it is subjective. The same argumentation is valid for assessment of effects and risks. These steps are subjective as well and the output depends on the builder of the business case. The cost step of the method is objective, but depends on the scope of the project. The alternative selection step can be either objective or subjective, depending on the non-financial effects of alternatives and their weight. In case alternatives only differ financially from each other, the decision is made objectively; the most profitable alternative is selected. In case other, subjective variables play a role as well, it depends on the person making the decision; how much weight does he give to these variables. Development of the implementation plan is also subjective and depends on the developer. However, this step does not influence the selected alternative. To reduce the effects of human bias, it is preferable that the business case is made by an independent actor, to increase the objectivity of the business case.

During creation of the business case, one of the experienced difficulties was switching between abstraction levels. A business model is an abstract representation of an organization. Processes and products are on a more tactical or even operational organizational level. The outcome of comparing business models in the business case depends on choices made in organizationally lower abstraction levels, like the tactical and operational level. The distinction between a process or product business case, and a business model business case needs to be made. In the first case, focus is on cost and benefit comparison of the innovated process or product. In the second case, it is about choosing the best alternative way of how an innovated product or process affects the business model.

Furthermore, we found some empirical evidence supporting the “strategy – business model – tactical set” framework by Casadesus-Masanell & Ricart [10]. In hindsight, the case study is mostly a product innovation within the tactical set of the building association’s business model. Some minor changes were made in the business model. This made it hard to devote the business case to the business model, and forced us to include more operational aspects in the business case. This is not per se negative for the demonstration, the method, or the outcome of the business case, but the goal and focus of the designed method, is to objectively compare two business models, in contrast with assessing the costs and benefits of a product innovation.

A limitation of the research is due to an almost complete lack of academic literature about business cases. The concept is used often, but without a well-designed and widely accepted methodology. As well as for the business model concept, it would have been better if a general accepted business case development method would have existed in academic literature for the reliability thoroughness of the research.

Overall, the method does what it is designed for. It is a method to develop a business case, which allows different business models to be compared, and the best one to be chosen as objective as possible.

7 Conclusion

The designed business case method to objectively compare business models can be used to compare and choose the best business model successfully, as demonstrated by the case study. The goal of this research was to increase the quality of the decision making process between possible business models, by developing a method to objectively compare the alternatives. Based on literature research, the business case method was designed. This method contains the eight components that Table 1 lists.

The need for a method like this comes from the increasing popularity of business models over the last decennia in practice as well as in academic literature. As discussed in the literature overview, companies that are aware of their business model performed significantly better then companies who are not aware of it [1], [13], [14]. Not only is the concept used more, but it also seems to increase organizational performance. Because a business model is an abstract description of a company, it is affected if the company changes. Whether on strategic level, business model level, or process and product level, innovation changes the company, and the business model. Most organizational changes can be modelled variously in the business model, each with a specific effect on the organization. Instead of just choosing one business model, a method is needed to compare the business models, and choose the best in terms of costs, benefits, and risks. The objective of this research was therefore to investigate the possibilities of the use of a business case as a method to compare business models, with the goal to define a method that increases the quality of business model decision making.

The case study showed that the developed method can be used to compare business models and choose the best one. However, the output of the business case depends partially on the people making the business case. Steps 3, 4, 5, and 7 are relatively subjective steps, which gives freedom to decision makers. Further research is needed to establish the effects of this decision freedom on the quality of the outcome of the business case. Still, the method fulfils the defined goal of the research.

References

[1] Chesbrough, H.: Business model innovation: It's not just about technology anymore. Strategy & Leadership 35(6), 12–17 (2007)

[2] Developing a business case: Expert solutions to oeveryday challenges. Harvard Business Review Press, Boston (2010)

[3] Meertens, L.O., Iacob, M.-E., Nieuwenhuis, L.(B.) J.M.: A Method for Business Model Development. In: Shishkov, B. (ed.) BMSD 2011. LNBIP, vol. 109, pp. 113–129. Springer, Heidelberg (2012)
[4] Peffers, K., Tuunanen, T., Rothenberger, M.A., Chatterjee, S.: A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems 24(3), 45–77 (2007)
[5] Wolfswinkel, J.F., Furtmueller, E., Wilderom, C.P.M.: Using grounded theory as a method for rigorously reviewing literature. Eur. J. Inf. Syst. 22(1), 45–55 (2013)
[6] Ward, J.: Building better business cases for IT investments. The Open University (2007)
[7] Remenyi, D.: Stop IT project failures. Routledge (1999)
[8] Van Ittersum, K., Pennings, J.M.E., Wansink, B., Van Trijp, H.C.M.: A multidimensional approach to measuring attribute importance. Advances in Consumer Research 31, 86–87 (2004)
[9] Centraal Fonds Volkshuisvesting, “Corporatie in Perspectief,” Baarn, The Netherlands (2012)
[10] Casadesus-Masanell, R., Ricart, J.E.: From Strategy to Business Models and onto Tactics. Long Range Planning 43(2-3), 195–215 (2010)
[11] Tah, J.H.M., Carr, V.: A proposal for construction project risk assessment using fuzzy logic. Construction Management and Economics 18(4), 491–500
[12] Kanji, G.K.: Implementation and pitfalls of total quality management. Total Quality Management 7(3), 331–343 (1996)
[13] Doz, Y.L., Kosonen, M.: Embedding Strategic Agility: A Leadership Agenda for Accelerating Business Model Renewal. Long Range Planning 43(2-3), 370–382 (2010)
[14] Shafer, S.M., Smith, H.J., Linder, J.C.: The power of business models. Business Horizons 48(3), 199–207 (2005)

The Business of Business Modeling

Coen Suurmond

RBK Group, Keulenstraat 18,
7418 ET Deventer, Netherlands
csuurmond@RBK.nl

Abstract. The research problem in this case study is to test a business focused approach to business process modelling as a foundation for the specification of an information architecture. The approach should be based on a common background and a good mutual understanding between practitioners from the business and the analyst, both for the modelling process itself and as a precondition for a valid model. In the case study some guidelines are explored to find the essential process model reflecting the general specifics of the business sector and the position of the individual company in the market.

Keywords: Organisational Semiotics, Business Modelling, Information Architecture.

1 Introduction

Subject of this paper is the modelling of the business processes of an enterprise (here named AYS) as well as the subsequent design of the structure of a new enterprise information system for this enterprise. The scope of the research consists of the initial stages of the development of completely new enterprise information system.

The research problem is twofold: to test a business focused approach to business process modelling and the specification of an information architecture, and to find some guidelines to separate the wheat from the chaff. Modelling means abstracting, and we need guidelines in modelling to decide what to include and what to leave out. Two kinds of business problems will be addressed in this case study. Firstly, there are the problems related to the validity of the models of the business processes. At the beginning of a project, IS experts and subject matter experts all have their own background, concepts, and language. This gap has to be bridged in order to communicate fruitfully. In the end, when the modelling of the business processes is finished, the question is whether the subject matter experts can successfully assess the validity of the created models. Do they adequately reflect the actual processes of this company?

Secondly, there are the problems related to the possible developmental trajectories of the company. The possible trajectories could be induced either by internal factors (e.g. the strategy of the company) or by external factors (e.g. requirements by stakeholders). The structure of the new enterprise information system should be able to cope with the foreseeable future scenarios of the enterprise.

B. Shishkov (Ed.): BMSD 2013, LNBIP 173, pp. 64–83, 2014.

The approach to both of these problems is to focus on the analysis of the ‘deep structures’ of the business processes of the enterprise, by analogy with the deep structures of grammar as defined by Chomsky. The surface structure of the business processes reflects the developmental history and context of the enterprise, and shows many idiosyncratic elements. The deep structure of the business processes, on the other hand, reflects the process logic of the processes, and will be the same regardless of the specific enterprise. To borrow some old terms from Aristotle: the process logic involves the essential characteristics of the business processes, the actual business processes contain the accidental characteristics, and history explains how the process logic is transformed in the actual business processes.

Take note: this partitioning in essential and accidental characteristics pertains to the modelling of the business processes. For the internal functioning of the enterprise as such, as well as for its individual success in the market, the idiosyncratic characteristics of the enterprise are the crucial ones. Every enterprise within a business sector will obviously have to meet general demands that stem from being in the market. However, its individual success as an enterprise is determined by how the enterprise adjusts its processes to fit the requirements and complexities imposed by its environment. A main risk for the deployment of ERP systems is to deny these specific characteristics of the enterprise (catch phrase: “best practices”).

This approach to the development of an information system is grounded in organisational semiotics. This research performed in the tradition of organisational semiotics is based on a sociotechnical approach of information systems as well as on the notion that human action in an enterprise is determined by social norms. The practice of system development should thus focus on exposing those norms together with the responsibilities in an organisation. The original contribution provided by this paper is found in the analysis of the process logic of the business both as a means of constructing a common background for the external analyst and the practitioner as well as a way to find a stabile information architecture for the business.

In this paper the problem, together with the approach and some general aspects of modelling will be discussed first. Next the theories will be discussed. The company in the case study is then described in terms of the most relevant features, followed by a section which explores a number of themes and guidelines for modelling business processes and the specifying of an information architecture. The final section discusses the results and future work.

2 Problem

The research problem for this case study is the design of a solid foundation for a newly to be developed information system for an enterprise by an external IS specialist, not familiar to the business or the enterprise. The practical aims for the enterprise concerned are (1) to be able to replace the current software package in the short term without loss of key functionality, (2) to expand the new system in the slightly longer term to provide the desired support for the enterprise’s business processes and (3) for the system to be capable of supporting possible strategic scenarios (of which it cannot be determined in advance if and when they will occur) at some later date.

The main idea behind the case study is that the stability of the desired information architecture is determined by its autonomy from chance factors and passing circumstances. In other words, the main idea is the notion that the essential and durable structure of the business processes should form the foundation of the information architecture. This introduces the question how this stability can best be found. This approach presumes that the characteristics of markets and products determine the essential structure of the business processes for an enterprise. To be active in a certain market, the enterprise has to follow a number of social, economic and legal conventions that are associated with the market and that place norms on the behaviour of the individual enterprise in the market. The same holds for the products of an enterprise, for both material and immaterial products. Of course, for material products a number of physical rules and constraints apply as well, such as food safety requirements in case of food products.

The idea is that hard statements can be made regarding the structure of the business processes and the associated information flows based on knowledge of the norms that apply for markets and products.

An additional motive to start the analysis of the structure of business processes with the markets and products is that this provides a better foundation for the collaboration between analyst and practitioners than the analysis of the current business processes of the enterprise. This will be explored further in a later paragraph.

3 Approach

This case study is an example of action research. Evert Gummesson [1] distinguishes societal action science and management action science. The former is based on a social and political view, while the latter focuses on the business of a company. Gummesson discusses ten points in relation to his concept of management action science. Most of these points concern the role of the researcher: he is a change agent in the business, the researcher has dual goals (obtaining results both for the business and for the research), and it is by definition an interactive and interpretative process aimed at understanding the business as a whole ("being holistic, recognizing complexities"). As far as data-generating methods are concerned Gummesson states that all kinds of methods can be used and that "qualitative, informal, in-depth interviews and the ethnographic methods of observation and participation are also important as part of action science". This case study exhibits the characteristics described by Gummesson.

The "preunderstanding of the corporate environment and the conditions of business" mentioned by Gummesson as an essential characteristic of management action science is indeed vital The researcher does not need to be fully aware of the ins and outs of the specific business at hand. However, a common view shared between the researcher and the participants from the business upon the business environment is required. The main characteristic of business in general is that markets and products take priority. If this is not accepted as the primary norm, analyses can easily head off in the wrong direction. It is exactly this understanding that allows the researcher to inquire about all kinds of actual practices and processes in relation to their contribution to product and customer value, as a mechanism to expose the underlying structures.

For the outcome of this case study there are two kinds of criteria: has it been of use to the business (what has changed as a result) and has it contributed to Information Systems Research as a scientific discipline. The use of the business will be evaluated by the participants from the business, both immediately after the analysis is done and after some time has passed. To evaluate the scientific results a distinction can be made between results in the area of conceptualising business processes and information systems and in the area of the applied method. To judge the results concerning conceptualisation the remarks made by Jean Perrin in his Nobel Lecture provide a good guide: "...and also because coherent assumptions on what is still invisible may increase our understanding of the visible" [2]. The hypothesis is that idiosyncratic concrete business processes in a specific enterprise can be understood as a superstructure built on stable deep structures of business processes belonging to the type of business.

3.1 Earlier Work

For the case study I will partly rely on my earlier theoretical work, as presented in ICEIS 2010 (unpublished paper), in BMSD 2011 [3] and in BMSD 2012 [4]; and on a long-term involvement with the Organisational Semiotics Community. I will also rely on long-term experience in the design of information systems for the food processing industry. However, this case concerns an electro-technical reparation enterprise and thus presents an interesting case for the transfer of practical experience between two very different lines of business.

4 How to Model, What to Include, What to Exclude?

The aim of the enterprise within this case is clear: replacement of the existing aged information system. The replacing information system must have an internal structure that can increasingly cover functionality in the business processes and that can accommodate possible new developments within the business and any associated new business processes. To build the desired information system, with the internal structure to support the indicated extensions and with the functionality to replace the existing system, the structure of the business processes needs to be modelled first. There is also a need to establish an information architecture that can handle the planned later developments and which is based on the process model with its tasks and the responsibilities that accompany the roles in this model. A practical question is how to arrive at this process model and information architecture?

The existing literature regarding the modelling of business processes and information architectures seems to fall short of providing sufficient support for several reasons. Firstly, the focus in process modelling research is very much on techniques and specification languages, graphical or otherwise. In other words, lots of research is performed regarding the technique of modelling and the visualisation of these models, with all kinds of consistency checks. However, very little research seems to be available regarding actual guidelines to be used in the modelling of business processes. Secondly, the research in information architectures is strongly geared towards the ICT components of the information flows. When you consider Mintzberg's definition [5] of the structure of organisations ("the sum total of the ways in which it divides its labor into distinct tasks and then achieves coordination between them") and the five accompanying coordination mechanisms that Mintzberg names (mutual adjustment, direct supervision, and standardisation of work processes, of outputs and of skills) it is clear that formalised information flows via computer systems can never cover the complete information supply of an organisation. In the direct contact between employees and in organisational routines (that are partly created because of standardisation) all kinds of information are hidden from the ICT approach. And in the physical world of an enterprise all kinds of information can be found that influence the execution of the processes. This can vary from posters on the wall and labels on the products to visual information obtained by viewing the products, storage spaces, production processes, et cetera.

The aims of the enterprise in this case require guidelines to analyse the business processes and to abstract them to the desired model along with the accompanying information architecture. To validate the process model and information architecture the practitioners of the enterprise have to be able to comprehend them. This means that they (1) can compare the models concerned with their current and future practice and that they (2) can indicate whether these models are an accurate representation of their business processes. To arrive at these necessary guidelines for modelling the processes and the information architecture I have gone back to some work predating the field of Information Systems Research. Firstly, we have the Aristotelian distinction between essential and accidental characteristics [6]. When an essential characteristic is removed from an object (or process), then this object (or process) ceases to exist as such. In contrast, accidental characteristics can be removed or added arbitrarily without affecting the nature of the object (or process). The second source is the distinction made by Noam Chomsky in linguistics between the deep structure and the surface structure [7]. As Chomsky puts it: The syntactic component specifies an infinite set of abstract formal objects, each of which incorporates all information relevant to a single interpretation of a particular sentence" and "The phonological component of a grammar determines the phonetic form of a sentence generated by the syntactic rules". Alongside these formulations by Chomsky it could be noted that the essential elements can all be found in the syntactic component while the accidental elements can be found in the phonological component.

It is, however, of the utmost importance (let us avoid the word essential for the moment) to analyse more closely to what extent this analogy is applicable to the research field in practice. Firstly, there is the dynamic and evolutionary nature of social reality. This is in contrast with the way both Chomsky and Aristotle considered reality in terms of predefined, timeless and inevitable characteristics. As a philosopher, Aristotle was searching both for essential and universal truths. Chomsky assumes an innate and universal human competence for language. In social science, we are searching for lasting, but non-universal, constitutional rules that define social realities. To take part in social interaction, one has to obey the constitutional rules.

In this sense they are essential rules. As social realities evolve over time, however, the constitutional rules might evolve as well.

Secondly, it is tempting to think that social bonds revolve primarily about the essential characteristics as described above. However, it is the pragmatic aspect, action itself, what it is really about. In spoken language an inflection of the voice can cause a sentence to take a radically different meaning, and mentioning a number of accurate facts in succession can suggest an inaccurate causal relationship ("directly after I flipped the switch, the sun set"). In enterprises we are dealing with social action in a given context as well. And the way in which its employees are willing and capable to take specific circumstances into account greatly determines the reputation of an enterprise. Of course, this does not mean that the underlying structure of essential characteristics is unimportant. It just means that this underlying structure does not determine the behaviour.

In his Allgemeine Modelltheorie Herbert Stachowiak provides three constitutional components of any model [8]. Firstly, a model stands for something else ("Abbildungsmerkmal", representation). Secondly, a model strips some elements of the thing it represents ("Verkürzungsmerkmal", leaving out, literally: shortening). Thirdly, models perform a certain function for specific subjects in a specific period in order to perform specific tasks ("pragmatisches Merkmal", commitment to action).

In modelling the business processes and the information architecture, the resulting models have to: show the deep structure of the processes (representation); show just the essential characteristics of the processes (shortening); and be suitable as a basis to realise the core of an information system that remains stable over an extended period (action). What the intended models definitely must not be: models for the "surface" of the information system such as it will be actually used by the employees of the organisation; and prescriptions for how the organisation ought to behave.

The models conceptualise the deep structures of the business processes. This implies that different organisations can set up their business processes in very different ways based on identical deep structures. The manner in which individual enterprises do so, and especially how the actual processes reflect the specific environment of the enterprise, are determining factors in the competitiveness of the enterprise.

5 Theory

5.1 Information Science

The focus in Information Systems Development and Information Systems Research is in general very much on the IT perspective. This is expressed for example in the development of new sub-disciplines within Information Systems Research which deal with Systems Architecture, Information Architecture and Enterprise Architecture respectively. Based on the names they were given it might be expected that Information Architecture and Enterprise Architecture would have the business rather than the ICT as its starting point. However, practice has proven differently. It would seem that the ICT world considers the failures of ICT systems to be the result of irregular behaviour of organisations and that is the ICT world that comes to the rescue by modelling information flows, business processes and enterprise structures as well as by the standardisation and rationalisation of processes. There is an implicit assumption that a well-designed and complete ICT system is both logical and consistent; the outside world may not yet be ready for it, and is thus in need of assistance to become better prepared to enjoy the benefits of ICT. In this Theory of Communicative Action Habermas talked about the colonisation of the lifeworld by the instrumental rationality of systems [9]. The Information System Research approaches exhibit many examples of this phenomenon.

Merlin Donald writes [10]: "Human cultures are unique in their cognitive nature: ideas and memories can be traded and shared among the members of a group" (Donald 2012, in Action, Perception and the Brain). The ideas and memories that were shared and exchanged before form the background against which new information is evaluated and interpreted. These processes happen continuously in each individual person form early childhood. When a new employee is introduced into an organisation or when he or she is moved to a different position within the same company, the new employee has to "grow into" the new social group. In the brilliant chapter A very good example of a socially well embedded information system (without computers!) is described by Checkland and Howell in their brilliant chapter "The Information System which Won the War" [11].

Within Information Systems Research there are a number of approaches that take into account the social nature of the organisation and the socio-technical nature of the information systems. Examples of this are provided by Language Action Perspective 12], Soft Systems Methodology (mentioned above) [11], the Demo approach [13], the communicative approach by Taylor and Van Every [14], and Organisational Semiotics 15, 16]. Each of these approaches emphasises the social and rule-based nature of any organisation. The social lifeworld of the employees, the norms and responsibilities accompanying the roles and tasks of the individual employee, and the dynamical nature of communication, language and meaning are all features that can be identified in each of the approaches mentioned.

However, two dangers are lurking in socio-technical approaches of information systems. Firstly, there is the risk of an assumed consistency and hierarchy in the outside world. For example, this seems to be the case with Stamper as founder of Organisational Semiotics when he models norms in a hierarchical system using society as its root. This ignores the crucial element of the functioning of employees in an organisation in which they regularly have to weigh conflicting norms against each other and in which they need to have the relevant information available to do so. An ICT developer might ask here for the weighting factors so that the system can calculate the outcome, which would be a denial of the contextual nature of such decisions. A further problem is the dynamic character of norms and its interpretations, they evolve over time. This is a normal evolutionary process, often only noticed when employees are asked to defend the choices they made in atypical situations.

The second risk in the socio-technical approach is that everything is subject to individual perspective and that grip is lost. When everyone has its own way of experiencing his own lifeworld, then how can we talk of an authoritative interpretation and steering behaviour? Normally, this is not a problem either in society or in organisations because there are a number of authoritative social norms present in the background. An isolated individual or group can ignore certain norms for an extended period and some norms might be set aside for a short while if there is majority support to do so, but stable societies and organisations are exactly those that are governed by stable underling norms (to paraphrase "you can fool some people all of the time, you can fool all people some of the time, but you cannot fool all people all of the time": some people can neglect fundamental norms all of the time, all people can neglect fundamental rules some of the time, but all people cannot ignore fundamental rules all of the time).

5.2 Theory of the Firm

An enterprise derives its existence from successfully delivering products to its markets. The two basic requirements for sustainable business are market demand and efficiency of production. According to Kay, every successful enterprise also has a form of 'uniqueness' that distinguishes it from its competitors and that cannot be copied [17]. This unique and idiosyncratic character of an enterprise determines its place on the market and can be found in partly intangible factors such as company culture, history and market trust or reputation. These factors can indirectly be found in the company culture and directly in the way in which individual employees are dealing with individual cases in the business processes. The latter is subject to acculturation processes, with conscious and unconscious, designed and historically grown mechanisms by which individuals learn "how things are done here".

This approach to the enterprise indicates that how an enterprise operates and the operations within an enterprise always have to be evaluated in light of its position in the market. This does not mean that the contribution to the market position is the only norm; there are inescapable human and societal norms after all. It does highlight that it is essential for the continuity of the enterprise that the market is the ultimate standard against which it is evaluated. This holds for operational actions and it holds as well for the actions taken by its management and for its strategic choices. Therefore, in analysing business processes and in designing an enterprise information system to support those business processes the orientation on the markets and products of the enterprise should be the first criterion.

From the above considerations it follows that the metaphor of the enterprise as an organism is more appropriate than the rationalistic and mechanistic approach of the enterprise, as described by De Geus [18]. After all, an enterprise is a social phenomenon in which the actions are determined by social norms and by interpretation processes. This means that modelling business processes and information flows from a purely rationalistic-mechanistic view or weakening the strengths of an enterprise by reducing the number of possible solutions in the business processes have to be avoided in the development of an enterprise information system.

5.3 Semiotics

Social communication happens through sign systems and the interpretation of signs is partly determined by history (the way in which signs were interpreted in the past) and partly by context (and sometimes by the way in which they are uttered as well, a certain inflection of the voice for example).

Within business processes the efficiency requires that much of the information can be processed by systems. The sign systems created to this end are of a formal nature: the meaning of variables and of possibly of value ranges is recorded in the systems in advance.

Within an organisation all kinds of capacities in which information can appear can be distinguished. Part of the information can be found in computer based systems, part is 'between the ears' by training, knowledge and experience and part is exchanged through all manners of ad hoc communication. The nature of the sign system determines the possible interpretations of the information given, in part because of the degree of formalisation.

Although semiotics remains in the background in the case study, semiotic insights certainly play their part in the analysis and modelling. This is especially evident in the meaning of sign systems and of interpretation processes in both the analyses and in the business processes. It is also visible in the prominent role played by social norms, in particular in understanding business processes against the background of the normative function of the markets and products of the enterprise [15, 16].

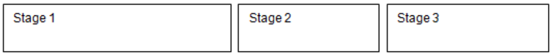
5.4 Process Modelling and Information System Development

Modelling business processes with the associated information flows, and validating the resulting model, is a communal activity of two different kinds of actors, each with a completely different background. On the one hand there is the analyst with communicative, analytic and modelling competencies (accustomed to formalised sign systems), on the other there is the practitioner with a detailed knowledge of what happens in practice, of organisational structures and procedures and equipped with lots of tacit knowledge.

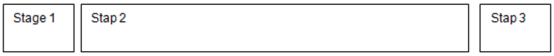
The difference in perception and background of the different actors cannot be bridged by the analyst transforming himself into a practitioner (or vice versa). As well as the time such a transformation would cost, it would mean a fundamental lack of recognition for the difference between the role of the analyst and the role of the practitioner. It might seem tempting to unite all of the required knowledge and experience in one person, but it would imply a major risk of consigning the process of modelling and analysis to the realm of tacit knowledge, with pernicious consequences for the validation and maintenance of the model. In effect it would be a one man show.

The model that is to be constructed of the business processes and the accompanying information flows should represent the essential structure, thus forming a stable basis for the information system that is to be developed. As a model it is an abstraction and not 'true' or 'false', but suitable to a greater or lesser degree for the purposes for which it was developed. A basic condition is stability: it should be possible to support all kinds of variations of the business processes by one single model. A second condition is the reduction of complexity: the model should enable insight into the complex reality of concrete business processes by omitting all kinds of details that are irrelevant to the structure and by naming the separate elements of the processes.

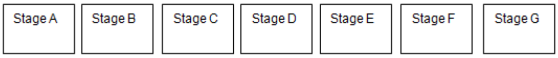
An abstract example of one aspect of modelling: say that a certain production process moves through three different steps and that these steps are modelled as they are observed in practice:



Later, the process is changed and the model with it:



However, if the process elements had been analysed further, the following model could have been the outcome:



In this last model with stages A through G both process variants could have been represented. Before the changes to the process stages A, B and C form the first sub-process, D and E the second and finally F and G the third sub-process. After the changes the sub-processes encompass different stages (A, B through F, G), but the basic model remains the same.

The major challenge is to distil those elements A through G from the concrete business processes with all of their details. It is not unusual to start from interviews with practitioners from different layers of the organisation combined with the analyst's own observations of the processes and information products. Often, documents regarding the organisation and those of the processes regarding quality control are important sources in gaining an understanding of the processes. However, in practice this springs a number of problems. The first issue is the degree to which the formal documentation of the organisation and its processes agrees with the organisational reality. Giving prominence to these documents implies taking a position regarding the value of these documents, either negatively ('worthless paper truths of the managers') or positively ('we are trying to act in this way, but it was not possible just now'). In both cases the formal documentation is the leading norm in taking stock of and evaluating the processes.

The second issue is the effectiveness and efficiency of the Interviews with practitioners. On the one hand the analyst can drown in details; on the other essential elements of the business processes might remain undiscussed. The analyst does not know they are there, while to the practitioner they are so obvious that it does not occur to him to mention them. The same holds for looking into the information products. How does the analyst find out what is not there, what is left out because it is supposed to be known or because the information is obtained by other means?

Another approach is working from the underlying norms of the enterprise. This begins with an orientation on the markets and products of the enterprise. After all, the enterprise exists because it creates products for customers and this is given shape in the business processes. The organisation (and quality control) has to structure and stabilise the business processes, but that should happen to serve the higher purpose:

to effectively serve customers in an efficient manner. Needless to say, other essential norms apply that lie outside of the enterprise. Those are in part societal norms and in part norms from specific stakeholders such as regulations by the government (requirements for the financial accounts are a striking example).

The norms that are based on markets, products and external stakeholders are in general more stable and accessible than all the ins and outs of the business processes (especially when the analyst has to work his way through lots of details before isolating what is structural from what is irrelevant for his purpose). On top of this, possible developments in those norms are essential for the internal structure of an enterprise. Those can be developments as a consequence of strategic decisions by the enterprise or external developments that the enterprise has to follow if it wants to remain in the market. The model of the business processes should be capable of following those developments without major structural changes.

5.5 Ontology and Ideal Type

Through the process logic the essential and stable elements of business processes and information flows should be mapped. This aim can also be distinguished in several ontological approaches. Essentially, the process logic is used to define a small and specific universe of discourse along with the associated operations. Using a classification of Poli the process logic could be placed under the term formalised ontology: "...to find the proper formal codification for the constructs descriptively acquired..." [19], with the essential difference however, that the intended constructs are not obtained by means of "collection of such prima facie information on types of items either in some specific domain of analysis or in general" [19], but by a normative and critical analysis of the enterprise against its background of its products and markets.

The use by Dietz of the term ontology points in the same direction: "Our goal is to understand the essence of the construction and operation of complete systems, more specifically, of enterprises" [13]. In a very different time and against a very different background Max Weber was searching for a precise and consistent description of social patterns and their backgrounds in his main work Economy and Society: "In order to give a precise meaning to these terms, it is necessary for the sociologist to formulate pure ideal types of the corresponding forms of actions which in each case involve the highest possible degree of logical integration by virtue of their complete adequacy on the level of meaning" [20].

A marked difference between the ontology approach as used in ICT and the use of the concept of an ideal type of Weber is the way in which the resulting model is viewed. Is it a basic design to engineer the social world towards what it should be or is it an instrument to understand patterns of rule-based human action in a specific context? The thinking behind the former idea is formulated clearly by Dietz: "Contrary to many dissenting theories that have been advanced in the past century, organizations are artifacts. They are systems that are, and have to be, designed and engineered, like any other artifact" [13]. The latter is expressed by Weber in two ways, directly following the earlier quote: "…it is probably seldom if ever that a real phenomenon can be found that corresponds exactly to one of these ideally constructed pure types." [20] and "The more sharply and precisely the ideal type has been constructed, thus the more abstract and unrealistic in this sense it is, the better it is able to perform its functions in formulating terminology, classifications, and hypotheses" [20].

The approach of using the process logic as a means to arrive at an information architecture shares characteristics with both of the above approaches. The concept of process logic is based on a Weberian idealisation and it is based on an analysis of the underlying norms of human action. The information architecture that is based on process logic is an especially good example of organisational engineering: a formal and consistent model of the essence of the business processes in an organisation. However, since the organisation as a social phenomenon is anything but an engineered system, but rather an emergent system that is continuously changing itself, the information architecture is not a prescription to how the organisation ought to behave. It works in the reverse direction: when the organisation behaves and develops itself as described by Taylor and Van Every [14] and when the actions of the organisations are at the same time determined by a number of inescapable rules, then it has to be possible to represent those matters within the capricious daily organisation reality that are essential to the business in the information architecture.

6 Description of the Case

6.1 Introduction

AYS is a leading service and repair business for mainly audio-visual equipment of major brands that operate nationwide. The enterprise carries out both on-site and carry-in repairs and has a network of six branches for the on-site and smaller carry-in repairs that service the different regions of the country. Larger carry-in repairs are performed centrally. The main contract partner is a leading brand (represented by its national importer), AYS is a certified partner and carries out all repairs in the Netherlands for audio-visual equipment of this brand. AYS is also active on a smaller scale in the repairs of other brands and of other kinds of electrical consumer products.

The key elements of AYS are:

- Both on-site repairs and carry-in repairs of audio-visual consumer products
- National coverage with six regional branches
- Strong affiliation with a strong brand
- Around 100 employees

6.2 Structure of AYS

The legal structure and the structure of the business processes are rather different at first viewing. AYS presents itself to the outside world as one homogeneous company with a specific service package. There is also a strong centralisation in terms of management and strategy; the head office defines the corporate identity and determines how the business is conducted. Legally there are a number of different entities (each a separate legal person) on three levels:

- Level 0: The holding
- Level 1: The main office and multiple entities that are not involved in the servicing and repairs and that will not be discussed here
- Level 2: The regional branches

The main office encompasses a number of central services, the main workshop with reception desk for carry in service and it provides the on-site service in its region. The regional branches provide the on-site service in their regions and they have a limited workshop facility with a limited reception desk service. The regional branches are either full subsidiary companies or fully owned by an independent entrepreneur.

6.3 Contracts, Agreements, Commitments

Curiously, there is only a very limited use of formal SLA's. The affiliation between the importer and AYS has much more the nature of a relational contract in which the details of the mutual obligations are not described as much as it is based on trust, established practice and, most importantly, on the binding effect of the settlement of financial claims of work carried out by AYS that are accepted or declined by the importer depending on the circumstances (circumstances that are not always known to AYS). Here, it is clear that this is not a symmetrical relationship; it is the importer who leads the way, who determines how matters are handled both materially and financially. In practice, there are a multitude of agreements and expectations regarding the handling of repairs (turnaround times, success rates) and regarding the handling of the financial side. Current practice is mainly based on the knowledge and experience of a number of key figures in the AYS organization (which is both a weakness and a strength; a weakness because of the dependence on individuals, a strength because it is hard to reproduce and thus cannot easily be adopted by others outside of AYS).

For other products groups and brands the same pattern holds and the size of the contract partner is there too defining for the (a)symmetry of the relationship between AYS and its contract partner.

6.4 Strategy

AYS has a growing strategy in two directions. The first direction is diversifying the brands. Because there is a strong current dependence on one brand, AYS is investigating the possibilities to apply the current competencies for audio-visual consumer products to different brands. Potential new activities are not expected to require new kinds of processes. However, it is possible that agreements and interactions with new parties will take on new forms (but that also holds true in regard to the current clients).

The second direction is to use the competencies and the nationwide network for new activities, in particular services to professional users. Such activities are potentially the servicing of permanent audio-visual installations, both for companies to whom that is the core process (informing and/or advertising to its customers) and for companies where it more concerns internal presentation capabilities. A different option is to provide the entire handling of defect equipment for larger retailers (logistics service partner). Another possibility is the provision of installations of new audio-visual equipment for professional users. Currently, there are some small-scale activities in these directions and growth towards full scale services is a real possibility.

6.5 Stakeholders

In principle AYS is dealing with one, two or three external parties and with one or two internal parties for a single repair job. The external parties are the end user (usually a consumer but it can be a company), the direct supplier (big chains of nationwide operating retailers), and the importer as representative of the brand. The client of AYS is one of these three parties and the details of the preceding parties often have to be registered as well (the consumer has two preceding parties and the importer has none). The contractor is one of the AYS branches which can subcontract the work in whole or in part to another branch of AYS.

Each stakeholder has its own way of providing and requesting (or demanding) information and of tracking the work and handling the financial side. Moreover, these patterns are subject to unpredictable change. The use of references by the stakeholders is also erratic. Standards for dealing with warranty conditions and for the execution of work differ per stakeholder. Market and power relations determine who is in charge, and as a smaller party AYS usually has to comply with the demands and expectations of the (much) bigger clients. Here, logic and facts can sometimes be set aside. The flexibility with which AYS deals with these complex and rapidly changing practices is an essential factor for the internal costs and for successfully getting the remuneration for the performed work.

NB: The term 'customer' is difficult to apply in the case of AYS, because there are so many kinds. Because of this, the term will be avoided as much as possible.

7 Themes and Guidelines in Process Modelling and Information Architecture

7.1 Process Logic

In this paper the term process logic has been used to distinguish it from the idiosyncratic characteristics of the enterprise. Usage of this term was founded in a number of considerations. Firstly, it deals with a schematic representation of the inevitable structures within a certain line of business, valid within a specific social environment. One might say that these are the structures a student should be taught, while he does not yet know which specific company will employ him. Secondly, norms for completeness and consistency hold for this schematic representation. On the level of abstraction chosen, it should be capable of representing every scenario that arises in practice (a tall order and a real challenge!) and there should be no inconsistencies or ambiguities. This demands definitions of the elementary terms and a precise formulation of the underlying norms.

In applying a concrete model of process logic it is essential to realise that it is an instrument to represent situations and processes (description, and an instrument for analysis) and that it not intended to be used to dictate how processes and situations ought to be (prescription). At its core, the process logic is a formalised sign system to (1) gain an understanding of the processes in the analysis, (2) precisely formulate terms and rules and (3) describe an information architecture that because of its character forms the basis for later system development. At the same time, process logic has to help the enterprise avoid inconsistencies (for example by preventing the use of key terms such as "service order" to mean various things) and leave the enterprise to choose how it sets up and executes its processes. An enterprise with five experienced employees will have to organise itself very differently from an enterprise with 5 offices, each with 20 employees and 10 flex workers!

In the case at hand the attempt to uncover the process logic has worked well, both to establish a common background between the analyst and the practitioner and to arrive at precise definitions, rules and demarcations.

Process logic is an important element for a common background, because it is a shared search for the underlying structures. For the analyst a general orientation on the specific markets and products of the enterprise with its peculiarities combined with a general background and common sense is sufficient to play his part in the discussions. All kinds of details that are hard to understand for an outsider can be isolated in this stage and assigned to specific places in the structure, without first needing to be fully explored or understood. This approach also forces the practitioners to be explicit about what really matters.

The approach also clarified what actually happens in the current business processes, as the examples regarding the concepts of process steps and service order have shown above. This conceptualisation of current practice allows for a very precise and fitting way of modelling and monitoring the business processes and leads to a better understanding by the practitioners of their own processes.

7.2 Administrations, Identities, and References

One of the pillars of process logic is the concept of an administration. The definition of an administration given by Starreveld is: "The systematic collection, recording, processing and supply of information for purposes of the managing and functioning of a household and for purposes of the accountability thereof". When we combine this definition with the idea that an administration concerns one specific domain, it seems obvious to directly name the required administrations when process logic is specified. Here, it is important to note that administrations concern product data and not master data.

The first criterion to arrive at an adequate administration is a high degree of homogeneity and autonomy. It must be possible to view the data that are collected in one administration as a single coherent whole. Also, the direct interactions with and dependencies on other administrations have to be as few as possible. A second criterion is the responsibility for its management; each administration in an organisation should have a single person who carries responsibility for the correctness, timeliness en completeness of the data in it. This responsibility should ideally be located close to the primary process, to ensure that those responsible are in touch with the reality represented in the administration.

In the case at hand this mainly means that each branch has its own administration and carries full responsibility for it. For example, there is no central registration of orders and stocks, but each branch maintains its own administration in these areas. Incidentally, in this concrete case it does not mean that they are free to choose their own systems. Everyone uses the same system, but within it every branch has its own administration. Of course, in the presentation layer connections can be made across the different administrations to enable central monitoring of the processes. And the serial number administration in which the service history of individual devices is registered is an example of an administration that must necessarily be kept centrally because of the nature of the data and the interaction of these data with external systems.

For the development of the new information system the specified administrations are composed of two parts: firstly a basic structure with entities and secondly their internal and external identities. Of course, within the database a single entity has a single unique identity, but inside and outside of the organisation an entity might have many alternative identities. Think of the number of a service order for example: internal and external stakeholders can use all kinds of references for themselves and use their own reference to request or provide data. Another example of this mechanism is the serial number: at first viewing this is a unique number. In practice, this number is a unique number within a specific brand, product group or model. A serial number thus does not uniquely specify a single device or part while it is required to do so. The enterprise also has a need at times to refer uniquely to a part that does not have a serial number, which can be met by assigning it a particular number generated for this purpose. When the part is gone, the number is as well. Based on these considerations, it is prudent to primarily assign unique numbers generated by the enterprise itself to parts and devices and to consider the serial number on a device or part as an alias to arrive at the generated number. This system is always applicable and avoids the complicated composite identification that results from accepting the serial number as identifier.

From the very beginning, the structure of the administrations has to be erected along with the associated references inside and outside of the enterprise. Further dressing up and setting up of the administration with data relevant to the contents, further status information, et cetera, can be done afterwards, in parallel to the development of the applications that use these data.

7.3 Lean IT

The lean approach places a number of demands on the set-up of an enterprise information system. Positively formulated, the information system must contribute to the effectiveness and efficiency of the business processes and use the most appropriate means to do so. Negatively formulated, the system is not allowed to cause waste (e.g.: excess production of information), to place undesired limits upon the business processes and it may promote the autonomy of processes as long as this does not harm overall efficiency and effectiveness. Information from the system has to be reliable and relevant.

Put otherwise: employees have to keep being presented with information in the right way, feed back information themselves and have the freedom to make their own decisions within their domain. Two examples illustrate the application of these criteria: First, the registration of direct hours on service orders. From the management there was a strong desire to gain a detailed view of the usage of hours in the primary process. In computer systems nothing is simpler than granting this wish: registration per service order, per department, and per activity of time used. In everyday reality however, such a system leads to unusable information. Firstly, because there is a mismatch with the way in which the work is actually carried out. Secondly, because it results in an excess of registrations. Either the categories are too general and the registrations limited, or the categories are specific and the registrations time-consuming. In both cases the registrations will provide an unreliable view of reality. That is why we opted to start with registration per service order in just the repair department, where it is registered for each employee when he begins and ends with a service order and which activities he performed during this time (which does not provide the time per activity). In this way, insight is provided into the ratio of time spend on service orders to time spend on other activities. Insight is also provided into the cases in which a service order has been handled repeatedly, by whom and to perform which activities. These data provide the foreman with a measuring stick to monitor the performance of his crew and to pay additional attention to activities that seem to take up too much time.

A second example is insight in what tasks must be performed and which tasks might. The turnaround time of service orders is one of the most important parameters for the performance of the enterprise with regard to the various stakeholders. When norms are created for the turnaround time as a whole and for the turnaround time of each individual process step, it is possible for the system to directly show which service orders have to be handled on a particular day in a particular team and which other work remains to be done with what time remaining according to the norms. This allows the team to make optimal use of its capacity by handling the service orders that are the best fit at that moment for the current activities and available resources. Self-control instead of central control should result in a significant advance in efficiency here.

7.4 Protecting and Strengthening the Distinctiveness of the Company

The strength of the enterprise is two-fold: nationwide coverage and a strong bond with a strong brand with strong partners. The downside of this connection to market heavyweights is that they determine what the service conditions will be to a large extent, both regarding the fees and regarding the mutual information supply. In this sense we are dealing with strongly asymmetric relationships. On the other hand, AYS is able to relieve large market parties of work that those parties are much less well equipped to handle and to carry out this work to a high standard. In other words, the enterprise has a clear place in the market.

Current legacy information systems of AYS have been built or adapted based on concrete developments in and requests by the market partners. Because of the aging core system solutions have been added to applications that were not originally meant for such. The new system should improve the ability of the enterprise to react to market developments through improved insight into the actual course of the processes by sharply demarcating the various administrations. This improved information position should result in an improved bargaining position with both existing and new contract partners.

A second contribution to the strategic position of the enterprise is also based on pulling apart the core administrations. In this way the enterprise is enabled to develop activities other than just service orders for defective equipment. The potential of nationwide coverage with service vehicles can be utilised for other activities as well. The information architecture allows both developments to be introduced gradually to expand into new markets, without major, risky, investment.

Finally, the flexible legal and financial structure is a major advantage. The current diversity with both fully-owned branches and branches exploited by independent entrepreneurs allows for rapid change both in acquiring work and in subcontracting it. AYS can profile itself as a strong market party because it presents a unified face to the client (in corporate identity and in home visits) and orchestrates the orders, while the work may be carried out elsewhere. Separating the diversity of the legal and financial structures from the unity of AYS as a business actor is an important requirement for the information architecture.

8 Summary and Conclusions

The results of this case study lie in two distinct areas, namely business and science. For the business, the approach was a success: The practitioners were able to compare the approach with an earlier analysis carried out by the supplier of a COTS solution. This process analysis was carried out in preparation for the expected implementation of their solution. The supplier had a tried and trusted reputation within the business sector of the customer. In practice, however, there proved to be little analysis of the processes in practice, but more of a technical and software-centric description of processes of the customer, expressed in terms of the proposed solution. Added on to this analysis were points for which the solution would be unable to provide adequate support and which thus required additional customisations. This gap analysis was also carried out using the terminology of the supplier and their solution. In other words, the customer did not receive a process analysis, but only a projection of the solution upon their processes. The customer was also unable to evaluate the provided description because of a lack of a common background.

In contrast, the analysis performed in this case study provided the customer with a lot of insight into their own processes. The method of deep inquiry after the underlying process logic was exactly what forced the customer to regard their processes in a critical manner. This also provided valuable information about and insight into the value of historically grown patterns and responsibilities in their business processes. Additionally, the analysis provided a good foundation to base the next steps in the automation of the company on because it conceptualises the business processes and assigns organisational responsibilities sharply. Furthermore, the structure described was assessed by the participants from the business as suitable to accommodate future developments

For the research goals of the action research this case provided mixed results. The method of analysis of the process logic has withstood the test of practice. However, the form for the final product, the description of the process logic, has not yet been found. The description as a list of definitions, a specification of the administrations and a specification of references certainly forms a useful and testable foundation, but a more formalised form of core entities and their transitions would be desirable.

It seems that the approach of searching for the process logic in cooperation with practitioners and to specify an information architecture guided by the concept of administrations, using the principles of Lean IT and maintaining the distinctive capabilities as preconditions, can lead to good results in practice. It is also an approach that allows the analyst and practitioners to discuss and weigh up options fruitfully, as long as the general understanding of the business sector mentioned by Gummesson is present. The approach is consistent with a number of relevant theories in business economics, management science, semiotics and the social sciences. However, there still remains a long way to go before theory and practice are unified into a single consistent whole.

To conclude, this case study had an exploratory character, trying out several ideas regarding both the theoretical background and regarding the application of the ideas in a real world project. As such, it succeeded; in a short time good results were obtained (especially in comparison to a previous analysis of the same company carried out by a potential software supplier). To give the approach a strong theoretical foundation will require a good deal of further work.

References

1. Gummesson, E.: Qualitative Methods in Management Research, 2nd edn. Sage Publications Inc., Thousand Oaks (2000)
2. Perrin, J.: Nobel Lecture: Discontinuous Structure of Matter. Nobelprize.org. Nobel Media AB 2013 (1926), http://www.nobelprize.org/nobel_prizes/physics/laureates/1926/perrin-lecture.html (accessed on Web January 27, 2014)
3. Suurmond, C.: Administrations as Instruments for Dealing with Organizational Complexity. In: Shishkov, B. (ed.) BMSD 2011. LNBIP, vol. 109, pp. 130–146. Springer, Heidelberg (2012)
4. Suurmond, C.: Sign Systems, Information Systems, and Engineering. In: Shishkov, B. (ed.) BMSD 2012. LNBIP, vol. 142, pp. 82–101. Springer, Heidelberg (2013)
5. Mintzberg, H.: The Structuring of Organizations. Prentice Hall, Englewood Cliffs (1979)
6. Barnes, J. (ed.): The Complete Works of Aristotle. Princeton University Press, Princeton (1984)
7. Chomsky, N.: Aspects of the Theory of Syntax. MIT Press, Cambridge (1965)

8. Stachiowak, H.: Allgemeine Modelltheorie. Springer, Wien (1973)
9. Habermas, J.: The Theory of Communicative Action, 2 vols. Polity Press, Cambridge (1986)
10. Donald, M.: The Slow Process: A Hypothetical Coginitive Adaptation for Distributed Cognitive Networks. In: Schulkin, J. (ed.) Action, Perception and the Brain, pp. 25–42. Palgrave Macmillan, Basingstoke (2012)
11. Checkland, P., Holwell, S.: Information, Systems, and Information Systems. Wiley, Chichester (1998)
12. Weigand, H.: LAP: 10 years in retrospect (2005), `http://www.vits.org/konferenser/lap2005/Paper%201-LAP.pdf`
13. Dietz, J.L.G.: Enterprise Ontology. Springer. Berlin (2006)
14. Taylor, J.R., Van Every, E.J.: The Emergent Organization. Lawrence Erlbaum. Mahwah (2000)
15. Liu, K.: Semiotics in Information Systems Engineering. Cambridge University Press, Cambridge (2000)
16. Stamper, R.: Information as a Social Scioence. In: Falkenberg, E.D., Lyytinen, K., Verrijn-Stuart, A.A. (eds.) Information Systems Concepts: An Integrated Discipline Emerging, pp. 1–51. Kluwer Academic Publishers, Boston (1999)
17. Kay, J.: Foundations of Corporate Success. Oxford University Press, Oxford (1993)
18. De Geus, A.: The Living Company. Nicholas Brealey Publishing. London (1997)
19. Poli, R.: Ontology: The Classical Stance. In: Poli, R., Seibt, J. (eds.) Theory and Applications of Ontology – Philosophical Perspectives, pp. 1–22. Springer, Berlin (2010)
20. Weber, M.: Economy and Society. University of California Press. Berkely (1968)

Engineering Business Processes: Comparing Prescriptive Guidelines from EO and NSBP

Dieter Van Nuffel, Philip Huysmans, and Peter De Bruyn

Normalized Systems Institute, University of Antwerp,
Prinsstraat 13, B-2000 Antwerp, Belgium
{dieter.vannuffel,philip.huysmans,peter.debruyn}@uantwerp.be

Abstract. Both Enterprise Ontology and Normalized Systems can be considered as theories provide prescriptive guidelines to design systems. Enterprise Ontology explicitly focuses on the design of organizations as being social systems. Originally, Normalized Systems focused on the design of evolvable software systems. However, it has been shown that, building on the Normalized Systems design knowledge, prescriptions for other domains, such as business processes, can be proposed as well. This domain seems to overlap at least partially with the domain of Enterprise Ontology, which is used to establish claims concerning process design in various publications. However, both theories are based on completely different kernel theories. Therefore, this paper analyzes to which extent the guidelines proposed for the Normalized Systems Business Processes are consistent, complementing or conflicting with prescriptions from Enterprise Ontology. This analysis is complemented by means of a case study elaborated from both approaches. A consistent set of prescriptions could lead to a more integrated approach for designing integrated organizations, business processes and software systems.

Keywords: Normalized Systems, Enterprise Ontology, Business Process Modeling, Enterprise Engineering.

1 Introduction

The design of organizations and their components (e.g., structure, business processes, and software systems) is an important topic in both practical and scientific communities [1,2]. However, explicit design knowledge in these fields seems limited. For example, Mendling et al. argue that many theoretical frameworks are too abstract, and that more practically-oriented guidelines lack empirical and theoretical support [3]. As a result, design of organizational components is often considered as craftsmanship, rather than engineering.

The enterprise engineering paradigm introduces a set of prescriptive design theories which seek to remedy this issue [4]. It specifically mentions the β and ν theories as well-founded theories to guide design efforts. The ν-theory states that the design of a system is normalized when a change consists of a set of

B. Shishkov (Ed.): BMSD 2013, LNBIP 173, pp. 84–105, 2014.

elementary changes, so that every elementary change does not trigger combinatorial effects [4, p. 101]. Normalized Systems (NS) provides concrete guidelines and design patterns to obtain such normalization in software systems [5]. Based on this approach, normalization of business processes has been researched as well [6]. This resulted in a set of guidelines to design Normalized Systems Business Processes (NSBP). Both approaches formulate unambiguous and theoretically founded guidelines based on the single postulate of obtaining the systems theoretic concept of stability. As a result, an approach which resembles traditional engineering, rather than mere craftsmanship, arises.

The β-theory states that enterprise architecture should be defined as deliberate restriction of design freedom, addressing the function design, construction design, and implementation design of systems [4, p. 100]. For example, Enterprise Ontology (EO) prescribes how the construction design of an organization should be made [7]. EO prescribes a clear way of separating different abstraction levels to be considered in organizations (i.e., ontological, datalogical and infological) and a systematic recurring pattern to model the ontological level.

While the formulation of such theories has been demonstrated to further the field in practice, several issues remain. One important issue is the current lack of integration between specific methods, thereby integrating different theories [4]. This means that, within the enterprise engineering community, additional research is required towards an integrated method consisting of different prescriptive design theories. For example, both EO and NSBP seem to provide a similar kind of guidelines when used in practical projects, which could indicate that both approaches could complementary in use. However, a clear obstacle when aiming to apply both approaches simultaneously, is their difference in theoretical backgrounds and abstraction. Therefore, an in-depth analysis regarding the possible compatibility of the guidelines resulting from both approaches is required upfront. Such approach would investigate the extent to which these guidelines are (1) similar (i.e., *consistent*), (2) providing additional guidelines (i.e., *complementary*), or (3) contradicting one another (i.e., *conflicting*). This approach does not result in a *theoretical* analysis of EO and NS(BP). On the one hand, NSBP cannot be theoretically EO-compliant, since the distinction axiom is not adhered to: no separation of ontological, infological and datalogical concerns is made. On the other hand, EO has not been developed based on the concept of systems theoretic stability. Nevertheless, we are convinced that analyzing thee consistency, complementalness, or conflictation of their practical guidelines can contribute to an integrated use of both EO and NSBP in various projects.

In this paper, we fill first provide some basic background regarding both NS and EO in Section 2. Section 3 will outline our approach for comparing the guidelines resulting from both theories, which will be implemented in Section 4. An illustration by means of the Library case study will be presented in Section 5. Finally, we offer our discussion and conclusion in Sections 6 and 7, respectively.

2 Background

2.1 Normalized Systems

NS theory is aimed at studying how modular structures behave under change [5,8,9]. Initially, the theory was developed by studying change and evolvability at the software architecture level, by applying concepts such as stability and entropy. Indeed, software systems —regardless of the programming language or paradigm they are based on— can be considered as modular systems (e.g., using classes, objects, procedures, etcetera) [5]. Typical advantages of modular structures (such as complexity reduction, reuse and evolvability) were included in the ideal future outlook of software engineering as expressed in 1968 by McIlroy [10] while arguing for "*families of routines to be constructed on rational principles so that families fit together as building blocks*". However, the criteria for deciding on how to delineate such modular building blocks are not trivial [11]. In addition, the aspired evolvability often seems absent in reality. For example Lehman's Law of Increasing Complexity [12] states that, as software programs are changed, their complexity increases and structures deteriorates (unless additional effort is invested to avoid it), which obviously hampers evolvability. Aiming to provide such truly evolvable software architectures in based on solid principles, NS theory applies the concept of stability as defined in systems theory [13,14] to software architectures. This implies that a bounded input function should result in bounded output values, even as $T \rightarrow \infty$. In other words, stability demands that the impact of a change is only dependent on the nature of the change itself. If the amount of impacts is related to the size of the system, a *combinatorial effect* occurs. Research has shown that it is very difficult to prevent combinatorial effects when designing software architectures. More specifically, it has been proven that combinatorial effects are introduced each time one of four theorems is violated (i.e., separation of concerns, data version transparency, action version transparency and separation of states) [8,9].

Various studies have shown that combinatorial effects, hampering the evolvability of modular structures, do not occur solely on the level of software architectures [6,15]. On the business process level, it has been argued that business processes at their most basic level (i.e., the "elementary tasks and elementary sequencing and design of these tasks" [6]) can be designed as modular structures based on NS theorems. In this context, business processes have been compared to production lines [5]. In this analogy, a business process flow performs operations on instances of a specific *life cycle information object.* Although production lines may seem highly integrated, they are actually loosely coupled. Every single processing step requires the completion of the previous steps on that instance of a particular product, but it does not require any knowledge of the previous processing steps, nor of the subsequent steps. As a result, changes to individual processes or tasks do not impact other processes of tasks [6]. Put differently, no combinatorial effects occur. More generally, a business process which does not contain combinatorial effects is called a Normalized Systems Business Process (NSBP). In order to achieve such processes, a set of guidelines has been developed, which are based on the more fundamental theorems of NS. Together, these guidelines allow the design of business processes without introducing combinatorial effects.

2.2 Enterprise Ontology

EO provides an organizational theory [7] which is based on the Language-Action Perspective (LAP) [16]. Consequently, it considers an organization as a social system, where the essential actions are performed by actor roles [17,18,19]. The identification of essential components in an organization is important for the goal of this paper, since this background results in the claim that EO equally provides "a modular framework for business processes" [20, p. 1]. The EO theory consists of four axioms (i.e., the operation axiom, the transaction axiom, the composition axiom and the distinction axiom) [7,21]. These axioms allow to specify in more detail what is meant with the "modular construction of business processes" [20, p. 18]. Business processes are considered to consist of three levels of building blocks. A first type of building block (the atoms) refers to the individual acts performed by actors, as explained by the operation axiom. These atoms can be combined in higher-level building blocks (i.e., molecules), which represent the transactions as explained in the transaction axiom. Multiple transactions can be required to fulfill a certain service to a stakeholder. The collection of these transactions (i.e., a fiber) is then considered to be a business process.

Rather than merely defining business processes using EO concepts, various studies have focused on the design of business processes. For example, the main research question of the paper *Basic Notions Regarding Business Processes and Supporting Information Systems* is "how business processes can be understood in such a way that their continuous and concurrent (re)designing and (re)engineering can be performed more effectively than what is currently the case" [22]. Another example is the paper *Enhancing the Formal Foundations of BPMN by Enterprise Ontology*, which states 11 propositions which can be derived from EO axioms [23]. Based on the axioms, additional prescriptions for designing business processes are available. For example, the operational cycle [7, p. 163] states that an actor role needs to be added when a transaction cannot be performed in the same cycle of other transactions. Put differently, this implies that the executor actor of an enclosing transaction needs to be the initiator actor of an enclosed transaction (cf. the composition axiom). Consequently, EO prescribes that certain end-to-end processes which are often defined in practice (e.g., order-to-cash processes) need to be separated.

Various claims have been made that EO can indeed lead to better results when (re)designing processes. The abstractions discussed in the distinction axiom are claimed to be "a tremendous advantage for discussing business process optimization" [7, p. 183], [24]. Moreover, the dedicated model within the DEMO methodology to represent business processes (i.e., the process model) has been claimed to "facilitate the discussion about the redesign of business processes" [7, p. 183].

2.3 Is It Possible to Compare Both Theories?

We have discussed how both EO and NS claim to apply modularity concepts at the organizational level. However, modularity is a concept traditionally applied within technological domains. It has been argued nevertheless that modularity reasoning is relevant in other domains as well. Focusing on organizations, multiple elements can be considered. First, in terms of its generally associated benefits, organizations are often equally looking for the right amount of complexity, (loose) coupling and agility in their structures. Second, existing studies show the relevance and feasibility of analyzing organizational constructs —such as products, production processes and organizational structures— in terms of modularity related concepts (for an overview, see for instance [25,26]). Therefore, it makes sense to apply modularity concepts, reasoning and theories such as NS and EO at the organizational level. As our aim is to look at guidelines for designing business processes, it is even more interesting that some authors note the current lack of theoretically founded prescriptive principles for designing their modular structure [27].

Still, caution should be applied when comparing the EO and NS theory, since their intentional application domains vary greatly. NS theory originates from research on the evolvability of the modular structure of software architectures, while EO attempts to describe coordination in organizations [17,18,19]. Nevertheless, the Design Science paradigm argues that the application of theories of related fields is useful to make scientific progress [28,29]. Moreover, Winter and Albani claim that different design theories can be combined in certain projects [30]. Both the NS and EO theory have already been positioned in a Design Science research framework [31,30]. Comparing these frameworks indicates an important difference between both theories: EO builds on communication theories (i.e., the theory of communicative action [32], the language-action theory [21] and systemic ontology [18]) while NS builds on system theoretic and thermodynamic concepts such as stability and entropy.

Notwithstanding this clear difference in kernel theories, remarkable similarities between NS and EO have been discussed as well [15]. For example, the resemblances in explicit state handling have been discussed in-depth [15, p. 212]. Moreover, other attempts have been made to integrate NS and EO theory more directly [33,34,35]. It should be noted that in these efforts, an inductive approach based on concrete artifacts is used, which can be contrasted to a more theoretical approach. Similarly, this paper does not attempt to provide a theoretical comparison, but aims to compare *similar components* of both theories on an *overlapping domain*.

The similar components refer to the formulation of *prescriptive guidelines* regarding modular structures by both theories. In NS, such guidelines are referred to by stressing the determinism of design [6]. In EO, we find clear references to the importance of such guidelines in the definition of architecture, which is "the normative restriction of design freedom" [7]. The *overlapping domain* is the domain of business processes, which is clearly addressed in NSBP 2.1). While business processes are defined within EO as well, it should be noted that we interpret the prescriptions of EO not only on the ontological level. In any organization, the ontological models eventually need to be extended to include the infological and datalogical layers, and to specify an implementation. Implementation means "the particular subjects that fulfill the actor roles at a particular time, the particular way in which C-acts are performed, and the particular way in which P-acts are performed." Several publications focused on this subject, which shows that a design is obtained which is influenced by EO prescriptions, but which can no longer be considered to be a design of a social system by itself, or to be entirely on the ontological level. For example, we mention research to define use cases for information systems based on DEMO models [36]. This is in line with insights from the generic systems development process (GSDP) [7, p. 71], which states that a functional specification of an object system needs to be made based on the constructional model of the using system.

3 Approach

In order to compare the guidelines of EO and NSBP, four categories should be considered: (1) *Consistent*: guidelines from both NSBP and EO prescribe the same design; (2) *EO-ignorant*: an NSBP guideline which has no similar EO guideline; (3) *NSBP-ignorant*: an EO guideline which has no similar NSBP guideline; (4) *Conflicting*: a NSBP guideline, which prescribes a different design than an EO guideline, or vice versa. Certain guidelines are expected to be consistent, since both EO and NSBP consider business processes as modular structures, and propose guidelines to optimize their design. However, given the different kernel theories of both approaches, and their non-identical goals, certain conflicting guidelines could be identified. Moreover, neither EO or NSBP claim to be complete. The claim from Dietz that "we do not intend to claim that ... even the whole ψ-theory is a sufficient basis for achieving optimally performing enterprises" [7, p. 81] indicates the validity of the EO-ignorant category. The claim from Van Nuffel that NSBP guidelines are necessary, but not sufficient, indicates the validity of the NSBP-ignorant category. We will adopt the work of Van Nuffel as our starting point as it explicitly lists a set of 25 guidelines, whereas the guidelines from EO have not been formally consolidated in such list exhaustively enumerating all guidelines incorporated in the method. Further, given this starting point to determine for each guideline to which category it belongs, the NSBP-ignorant category will not be required in this paper.

The authors of this paper independently made a classification of the NSBP guidelines. After integrating the result, differences were discussed, and the assessment was iteratively refined. All three authors have a sufficient background in both EO and NSBP. The NSBP PhD dissertation [6] and EO book [7] were used as reference materials. Several academic publications were used for additional details. Moreover, several cases (see e.g., [6], [7], `http://www.demo.nl`) were consulted as an application of the guidelines.

4 Comparison

Within this section, the actual comparison between the practical guidelines resulting from the two theoretical approaches is made. Our discussion will follow the division made within the PhD of Van Nuffel [6]: first, the general guidelines with respect to identifying business processes are discussed. Second, the comparison continues with the three additional guidelines that in specific cases identify business processes. Third, the comparison continues with the guidelines determining individual tasks, and finally, the auxiliary guidelines are investigated. The business process patterns discussed in the PhD of Van Nuffel [6] focus on issues not discussed by EO, and are therefore not taken into account. This section lists the names of the guidelines in italic and bold font. Next, the guideline is summarized in italic. Then, the consistency, complementalness or conflict with EO is discussed. An overview of these discussions is provided in Table 3.

4.1 General Business Process Guidelines

1.1 Elementary Business Process: *A Business process denotes a constrained sequence — i.e., sequence, iteration or selection — of individual tasks representing state transitions in the life cycle of a single life cycle information object.* Within EO, a P-fact is a factum, which is defined as *"the result or the effect of an act"* [7, p. 42]. Therefore, facta "can be conceived as status changes of ... an object in some class" [7, p. 42]. Furthermore, the order in which facta occur is determined by so-called occurrence laws [7, p. 43]. The transaction is thus about a unique P-fact transcending the transaction pattern, which can be considered to be somewhat consistent with a NS business process which is about state transitions of a single life cycle information object as stated by NSBP. The one-to-one relationship between a transaction and a P-fact is in our opinion conceptually consistent with the one-to-one relationship between a single life cycle information object and a business process.

1.2 Elementary Life Cycle Information Object: *an information object not exhibiting state transparency is a life cycle information object.* Whereas NSBP prescribes the criterion of state transparency (i.e., when no proper state transitions should be made explicit [6, p. 118]) to define whether an information object is a genuine life cycle information object processed in a business process, EO does not explicitly state a rule, criterion or law that in all circumstances denotes what a single P-fact is. There are evidently ways and requirements a P-fact should adhere to, but no general identification mechanism seems to be made explicit:

- "We conceive the result of a production act as a particular change in the state of the system's object world" [7, p. 58];
- "The object world reflects the produced things (e.g., goods or services) that are delivered to the elements in the environment" [7, p. 58].

As a consequence, – although it could be argued that only most fine-grained production facts exist (and therefore, that production facts are defined unambiguously), but that they can be aggregated to simplify models – it seems that identification of production acts in EO is not unambiguous: it depends on what is considered to be the system and environment, and different production facts can be identified depending on the aggregation level taken into account. Moreover, elementary life cycle objects can also refer to infological and datalogical production facts. Therefore, the authors categorize this guideline as EO-ignorant.

1.3 Aggregated Business Process: *In order to represent an aggregated business process, an aggregated life cycle information object has to be introduced.* In EO, a business process is based on the composition axiom: "a business process is a collection of causally related transaction types, such that the starting step is either a request performed by an actor role in the environment or a request by an internal actor role to itself" [7, p. 103]. Based on this definition, the operational cycle [7, p. 163] can be understood, which specifies that certain end-to-end processes cannot be considered as causally related transactions. Since the NSBP guideline is explicitly aimed towards representing *any* required end-to-end process, both theories are conflicting in most situations.

1.4 Aggregation Level: *Tasks performed on a different aggregation level denote a separate business process.* Although in the PSD-diagrams the causal and conditional links are enriched with cardinalities that describe the relationship between different transactions, nowhere is indicated that when an analyst discovers an one-to-many relationship between two candidate transactions, both should be separated. Furthermore, this latter relates to the aggregation level on which production facts are defined, since a production fact defines a transaction. Again, this does not result in a guideline to actually separate the transactions. Therefore, EO seems to be ignorant with respect to this design issue.

1.5 Value Chain Phase: *The follow-up of an organizational artifact resulting from a value chain phase denotes a different business process.* While some arguments can be made for the consistency of this guideline, the most important argument seems to indicate a conflict. For example, the operation axiom might indicate value chain phases as separate transactions, although it is dependent on the aggregation level on which the P-facts are defined. Moreover, the composition axiom illustrates the possible nesting required to integrate the different phases. However, the transaction axiom results in design decisions like explicitly stating that the Order phase belongs to the Delivery process in a typical Customer Order process scope. With respect to the latter, NSBP clearly state that these phases should be separated as they denote separate concerns [6, p.132-34]. In this way, NSBP seems to consider concerns a level "deeper" as it explicitly considers a delivery not to belong to the Order Phase, but as a separate process in the aggregated business process Customer Order. Therefore, both theories seems to disagree with respect to this design issue.

1.6 Attribute Update Request: *A task sequence to update an attribute of a particular life cycle information object that is not part of its business process scenarios, is represented by an Attribute Update Request business process.* The guideline prescribes to separate state transitions dealing with modifying an attribute of a life cycle information object that does not belong to the business process scenarios (i.e., included in the process). EO however, considers such requests to change to be part of the transaction. Mostly, they can be represented by one of the four cancelation patterns. As such, this represents a conflict between the two theories, although they comply with each other on modifications that do belong to the business process scenarios.

1.7 Actor Business Process Responsibility: *Actor business process responsibility indicates a separate business process if different actors are responsible for a different set of tasks, of which the task allocation belongs to different process owners.* The operation axiom declares actor roles to denote chunks of authority, responsibility and competence. Furthermore, following EO, a single transaction can only be executed by a single actor role. In this way, this notion is equivalent to stating that state transitions of a particular life cycle information object being part of the responsibility of a particular process owner denote a separate business process. Furthermore, in addition to EO, also NSBP identifies the only vaguely described notion of process ownership within literature. As a consequence, NSBP opts for a clear identification of such process ownership, which seems to be very closely related to EO's notion of authority.

1.8 Notifying Stakeholders: *Because notifying, or communicating a message to, stakeholders constitutes an often recurring functionality in business processes, a designated business process will perform the required notification.* EO considers notifying stakeholders as performing coordination acts, which are part of an ontological transaction that creates a single P-fact. However, NSBP identifies the concern of notifying stakeholders to clearly differ from the concerns taken care of by other business process (e.g., delivering an order, recruiting an employee, etc.): "delivering a message in the correct format to the intended recipients at the right time in an unchanged format, with the related fault handling" [6, p.143]. These concerns refer to implementation details, which are not considered on the ontological level. Therefore, EO theory is ignorant with respect to this design guideline.

1.9 Payment: *Because paying a particular amount of money to a particular beneficiary constitutes an often recurring (technical) functionality in business processes, a designated (technical) business process will perform the required payment.* The payment business process/transaction is identified by both theories, and can be considered as consistent. Various DEMO cases illustrate this. It should be noted that NSBP requires that at least the execution phase of payment processes is implemented using a reusable business process, in order to prevent combinatorial effects. This is not clear from the DEMO cases, which explicitly define multiple payment transactions.

4.2 Business Process Guidelines

2.1 Product type: *A different type of product or service denotes a main concern, and thus indicates a different business process.* The composition axiom seems to indicate that EO also recognizes the existence of transactions that although being enclosed by the same transaction, do constitute individual and independent transactions based on a product structure. But again, no clear rules could be identified, implicitly stated by "one could apply a finer-grained product structure" [7, p. 170]. The notion of a product type defined by Van Nuffel [6, p. 149] allows some interpretation as well, namely the domain expert who will identify the characteristic dimensions on which product types exhibit similar properties. As a consequence, we categorize this design issue as an EO-ignorant one.

However, if the Logistics example discussed by NSBP is taken into account, the design issue also seems to indicate conflicting statements by the two theories. The NSBP separate the Logistics processes based on the following types: non-food, food, quickly rotating, slowly rotating, and so on. On the other hand, EO theory seems to declare that these product types do not cause another type of P-fact to be created, and thus no separate transaction to be executed. This could indicate a potential conflict.

2.2 Stakeholder type: *Stakeholder type should principally be considered a cross-functional concern (i.e., a concern which does not require a life cycle information object by itself), except for those business processes where the stakeholder type denotes the life cycle information object.* Whether the theories comply, comes down to the question: does EO consider a transaction to be independent from the actor role for which it is potentially performed? In the PhD of Van Nuffel, a case about Human Resources (HR) processes is discussed in which it is clearly demonstrated that the assignment processes for a statutory employee and a non-statutory employee differ. Based on the authors' knowledge, EO does not provide any rule or prescription about the potentially different nature of a transaction. For example, in the Educational Administration case, no separate transactions are created based on different student types.

2.3 Access Channel: *The concept of an access channel indicates a cross-functional concern.* In EO publications no explicit referral to this design question could be found. However, implementation is explicitly out of scope for EO: EO "fully abstracts from the implementation [of C-acts]", which includes "the particular way in which C-acts are performed" [7, p. 83]. Consequently, it can be argued that the theories comply as EO does not explicitly states a different access channel denotes a separate transaction. Consider in this context the pizzeria case [7, p. 166]. The transaction *T01: Completion* contains all access channels to place an order.

4.3 Task Guidelines

3.1 A Single Functional Task - Overview: *A task represents a functional entity of work that either results in a single state transition of a single information* *object type, or refers to an Update or Read task on a single information object type.* Where NSBP specifically describes what a single task (or step within a business process) can be, our analysis of the EO fails to find equivalent rules. Of course, the transaction axiom identifies single acts (resulting in facts) within the transaction which might indicate consistency. However, the authors seem to find more evidence to categorize it as EO-ignorant. For instance, consider the acceptance of a stated P-fact consisting of an evaluation of its quality by performing three quality tests and then communicating the outcome to the initiator actor role which is authorized, responsible and competent to accept the P-fact, who will communicate it to the executor. EO considers this example to be part of the Accept C-act whereas NSBP prescribes to separate it in five different tasks, and two instances of the Notification business process. Thus, based on our analysis, we consider it to be EO-ignorant.

3.2 CRUD Task: *Each of the Create - Read - Update - Delete (CRUD) operations constitutes a single task.* Since these tasks are on the infological and datalogical layers, this guidelines is EO-ignorant.

3.3 Manual Task: *Every manual task of which the initiation and completion has to be known, has to be designed as a separate task.* EO makes abstraction of the implementation of C- and P-acts (also see discussion of 2.3 Access Channel). Therefore, EO is ignorant with respect to this guideline.

3.4 Managing Time Constraint: *The management of a time constraint denotes a separate task because it represents the individual concern of managing a particular time constraint.* In EO, a time aspect only seems present in the time-aspect of the proposition of a P-fact [7, p. 84] and self-initiating transactions [7, p. 99]. However, EO makes no claims whatsoever with respect to (not) separating an individual time constraint. As such, we categorize the guideline to be EO-ignorant.

3.5 Business Rule Task: *A single business rule should be separated as a single task.* An individual business rule should be isolated in its designated task following NSBP. EO acknowledges that business rules can sometimes be existential laws, as expressed in the state model, or action rules, which are expressed in the action model [7, p. 196]. In this sense, both seem to be consistent. However, EO does not explicitly states that every single business rule should be isolated. Therefore, EO seems to be rather ignorant to this design issue.

3.6 Bridge Task: *When a business process instance operating on an instance of life cycle information object type I has to create a business process instance of another life cycle information object type L, this functionality is designed as a bridge task that initiates the creation of the instance of the life cycle information object L, and represents a state transition on the instance of I.* As already illustrated above, the composition axiom of EO denotes the nesting of transactions. As such, it is illustrated that the Request C-act can be “triggered” by another transaction (i.e., the executor of an enclosing transaction can initiate an enclosed transaction). The Result structure analysis step of the DEMO methodology also adds to this. Conceptually, this is what a bridge task represents: it triggers the execution of another business process.

3.7 Synchronization task: *When a business process instance operating on a life cycle information object I has to inform a business process instance of another life cycle information object L, a synchronization task, representing a state transition on the instance of I, alters the state of the business process instance of L.* The NSBP synchronization task conceptually equals the waiting conditions specified in the EO model based on the Result structure analysis, following the composition axiom.

3.8 Synchronizing Task: *A synchronizing task represents the task receiving information from another business process's execution, in order to continue the business process control flow.* Equivalent to the Bridge task, also the Accept C-act in the EO transaction pattern represents conceptually the same as a synchronizing task. It allows the enclosing transaction/business process to continue, and thus is the end of the waiting condition.

3.9 Actor Task Responsibility: *A task cannot consist of parts that are performed by different actors.* Here NSBP is consistent with EO, as the operation axiom states that actor roles are elementary chunks of authority, responsibility and competence. Thus the fact that another actor role is authorized, responsible and competent to perform a particular task, suffices to split this task from any other task another actor role is authorized, responsible, and competent to execute.

4.4 Auxiliary Guidelines

4.1 Unique State Labeling: *Each state of a life cycle information object has to be unique.* The first auxiliary guideline, *Unique State Labeling*, states that each state of a life cycle information object should be unique. Thus, it indicates the necessity to uniquely define the states a business process can transverse. Also EO identifies unique labels as each coordination act and each transaction are uniquely labeled; and even more it states that facts can be created, but cannot be undone [7, p. 82]. Thus theories are considered to be consistent.

4.2 Unique State Property: *A life cycle information object instance can only be in a single state at any time.* Also EO declares a transaction has a unique status: the last performed fact, which is defined in EO as a state transition in the C- or P-world [7, p. 82]. Thus theories are considered to be consistent.

4.3 Explicit Business Process End Point: *If a business process type has multiple possible outcomes, each of these scenarios should have its dedicated end* *point reflecting the respective end state of a business process instance.* EO specifies through its transaction patterns (basic-standard-cancelation) that every scenario should be explicitly described. In this way, it is consistent with NSBP as every business process' execution results in a specific end point/state, and not in a general state "finished".

4.4 Single Routing Logic: *A split/join element in a business process's control flow should only represent a single split or join routing expression.* Essentially EO does not discuss this proposed guideline, so it is considered to be EO-ignorant. However, it can be argued that both theories are consistent because within the transitions between the different C-facts and P-fact that are exhaustively described in the transaction pattern, no violation to the NSBP guideline was identified. Further research should identify whether this non-violation is purposefully – and thus the theories are consistent – or rather by chance – and thus remains EO-ignorant.

5 Case Study

In this section, we complement our general comparison of EO and NBSP guidelines from the previous section with an illustrative case study. The aim is to demonstrate the consistent, complementing and conflicting results when applying both approaches independently on the same case. The case used in this paper is the Library case, one of the two running case studies used in the book Enterprise Ontology [7, p. 218-220]. The scope of the library case can be summarized as follows:

A person can become a member of the library (because you have to be a member to borrow books). A member has to pay its membership fee. Some people (elderly, disabled people, etcetera) can apply for a reduced membership fee. A member can borrow a book from the library. A member cannot borrow more than five books simultaneously. A member can extend the loan of a particular book. A member can return a book. A member might have to pay a fine if she returns the book too late. Every month, the librarian decides which titles should be added to the catalogue and how many book copies should be ordered. Every year, the library sends out invoices to the current members.

The next sub-section contains the solution from the NSBP perspective. The second sub-section compares the two solutions which will be limited to comparing the different business processes identified by both approaches. Every year, the library sends out invoices to the current members

5.1 NSBP Case Solution

Within the case, two principal requirements consist of managing a member and a loan. Regarding a member, two main concerns are identified: registering a new member and collecting the yearly membership fee. The first clearly includes the creation of a new membership, thus `Membership` is the life cycle information object on which the *Create New Membership* business process operates. The second operates on a, rather artificial, life cycle information object `Yearly Fee Payment` that uniquely associates with the yearly fee payment of a single `Member`. This business process does not operate on the `Membership` because such a yearly fee payment only changes an attribute of the `Membership`: the attribute that states whether the yearly fee has been paid or not. Therefore, this business process complies with Attribute Update Request guideline.

The *Invoice* being prepared for each membership fee is considered to be a separate business process based on the guidelines Elementary Business Process and Elementary Lifecycle Information Object. An invoice clearly denotes something else than a membership fee, and furthermore, the concern of an Invoice is repeatedly encountered within a library: each time you have to pay a fee, a fine, etcetera. It denotes the activities required to generate an Invoice to the customer, which in a library might be limited by generating a document and linking it to the bookkeeping. Furthermore, an `'Invoice`' life cycle information object is not state transparent to any other information object encountered in the Library case. As a consequence, it also complies with the Elementary Lifecycle Information Object guideline. Finally, the invoicing also represents a phase that returns in almost all customer facing value chains. Also the concern of a single *Payment*, as defined in the theoretical section on NSBP, should be isolated in its designated business process.

Both *Member* business processes exhibit the possibility to apply for a reduced fee: when creating a new member and when collecting the yearly fee of the library's members, reduced fees can be attributed. The eligibility for a reduced fee has to be acknowledged every year: for instance, minors will get a reduced fee, but when attaining the age of majority, they should pay the full fee (except if they are students). This indicates that a `Reduced Fee Application` denotes a separate concern, because it recurs in at least two situations, indicated by an elementary life cycle information object `Reduced Fee Application`.

The second main requirement for a library consists of managing a loan. First, a `Loan` denotes that one `Member` has borrowed a particular `Book Copy` from the library: a `Book` has multiple `Book Copy` related to it. This reasoning is equivalent with making a distinction between the conceptual product, `Product Type`, and the physical product: `Product`. The main concern addressing this requirement is clearly the `Loan`, and thus constitutes a business process applying the Elementary Business Process guideline. A loan is created when a member borrows a book copy from the library, and ends when the member returns the book copy to the library.

Another design decision concerns the multiple interactions a `Member` can initiate with the library: she can loan one or more books, and/or she can extend the loan of one or more books, and/or she can return one or more books, and/or she can ask information at the counter. The latter is not considered to constitute a business process, but rather a task of the librarian to appropriately answer the question(s). A noteworthy observation is the use of the words "and/or" in the list of interactions, exhibiting that a `Member` might want to initiate multiple interactions, implying tasks — state transitions — on multiple `Loan` instances. Thus, a single member interaction relates in one-to-many way with managing a single `Loan`. In that sense, the Aggregation Level guideline implies a `Member Interaction` and managing a `Loan` to denote different concerns. Thus, if the library wants to explicitly monitor each `Member Interaction`, a *Member Interaction* business process has to be identified, illustrating the option to explicitly monitor each task sequence as a business process.

The third aspect includes some more management-oriented processes of a library. Every month, the librarian verifies which book titles and how many copies of them should be added to the catalogue. It is assumed that this management process consists of a number of activities like verifying the announcements of new books, controlling the remaining acquisition budget, and matching new titles with the current portfolio of books available in the library. This management process ultimately results in a number of book copies of a number of book titles to be ordered at a number of publishers. If it is assumed that the library joins the different book copies to be ordered at the same publisher, potentially multiple book orders result from the same catalogue management process. As a consequence, based on the guidelines Aggregated Business Process and Aggregation Level, the two concerns should be splitted in their individual business process running on the life cycle information object `Catalogue` and life cycle information object `Book Order` respectively. At this time, abstraction is being made from the fact that this book order can be partially delivered to the library, identifying the need for additional business processes. The same reasoning has to be applied with respect to the yearly fee management process: the concern of collecting the yearly fee for an individual member already discussed before and defined by the *Collect Yearly Fee* business process, clearly happens on a different aggregation level than the activities performed at the *Yearly Fee Management* business process that defines the overarching activities to be executed in order to collect the yearly fee of all members. For example, this businss process might include defining the yearly fee for the different membership types.

Finally, on several occasions the concern of `Notifications` can be identified within the Library case. Numerous messages are exchanged between the different actors. As discussed in Section 4, this concern requires its designated Notification business process having a rather fine-grained scope.

To conclude this subsection, Table 1 summarizes the business processes identified by the NSBP approach with a brief repetition of the rationale for their identification.

5.2 Comparison of the Case Solutions

The Library case solution by EO is extensively documented in [7], and is summarized here by its Transaction Result Table (TRT) in Table 2. When comparing Tables 1 and 2, a set of differences between the EO and NSBP case solutions can be found.

The first and main difference between both approaches is situated at the identification of business processes handling the *Loan* itself. EO prescribes another

Table 1. Library - Identified NSBP business processes within Library case study

Business Process	Manifestation of Guideline	Life Cycle Information Object
Create New Membership Process	Elementary business process & Elementary life cycle information object	Member
Collect Yearly Fee Process	Elementary business process & Elementary life cycle information object	Yearly Fee Payment
Reduced Fee Application Process	Elementary business process & Elementary life cycle information object	Reduced Fee Application
Member Interaction Process	Aggregation Level	Member Interaction
Loan Process	Elementary business process & Elementary life cycle information object	Loan
Invoice Process	Elementary business process, Elementary life cycle information object & Value Chain Phase	Invoice
Notification Process	Notifying Stakeholders	Notification
Payment Process	Payment	Payment
Catalogue Management Process	Aggregated business process	Catalogue
Book Order Process	Elementary business process & Aggregation Level	Book Order
Yearly Fee Management Process	Aggregated business process	Yearly Fee Management

Table 2. Library - Transaction Result Table [7, p.160]

Transaction Type	Result Type
T01 membership registration	R01 membership M has been started
T02 membership fee payment	R02 the fee for the membership M in year Y has been paid
T03 reduced fee approval	R03 the reduced fee for M in year Y is approved
T04 loan start	R04 loan L has been started
T05 book return	R05 book copy C has been returned
T06 loan end	R06 loan L has been ended
T07 return fine payment	R07 the late return fine fee for loan L has been paid
T08 book shipment	R08 shipment S has been performed
T09 stock control	R09 the stock control for month M has been done
T10 annual fee control	R10 the annual fee control for year Y has been done

design than NSBP: it separates the single *Loan* business process of NSBP into the three different interactions: T04 loan start, T05 book return, and T06 loan end. However, following NSBP, none of the identified guidelines, nor any other interpretation of the Normalized Systems principles would imply to split the *Loan* business process. First, the life cycle information object underlying all three mentioned interactions always refers to a single `Loan`, which clearly does not denote another, a collective, nor an aggregated life cycle information object. Second, the three identified interactions do not refer to another aggregation level, nor to distinct value chain phases. Third, the `Update Attribute Request` guideline does not apply to extending a `Loan`, because it constitutes a regular scenario of the *Loan* business process: each `Loan` instance might be extended zero or multiple times. As a consequence, extending a `Loan` does indicate a regular *Loan* business process scenario to update the `endDate` attribute of a `Loan` information object instance. Fourth, a different `Actor Process Responsibility` does not seem to apply, as most library clerks are eligible to handle all three identified transactions. Only deciding whether or not to grant a reduced fee belongs to a different responsibility.

The second difference is about the *Invoice* business process identified by NSBP. In EO, an invoice represents the request fact for performing a payment transaction. While such facts are identified at the ontological level, they are considered to be part of another transaction, and are not considered independent from such transaction. NSBP considers the Invoice process to denote a separate business process at it manages the activities necessary to create an Invoice. When analyzing the library case from the latter perspective, it becomes clear that the follow-up of an invoice is different from the other concerns (member, loan, payment, etcetera).

A third difference regards the inclusion of business processes in NSBP which, according to EO, belong to the infological and datalogical levels. As already set out in Section 4, EO does not state anything about the *Notification* business processes whereas NSBP strongly stresses the importance to split those kind of business processes. Given the definitive categorization of an invoice as a datalogical element by EO might suggest that concerns as defined by NSBP are cross-dimensional with respect to EO, i.e., when labelling the identified concerns by EO, it would result in ontological, infological and datalogical concerns.

A fourth difference consists of the *Member Interaction* business process which potentially might be identified based on the NSBP principles. From an EO perspective, these interactions are considered only in the implementation of the ontological actor roles. Therefore, no dedicated modeling artefacts will be included in the implementation-independent ontological models. Nevertheless, from a NSBP perspective, each concern an organization wants to monitor, should have its designated process and data element.

6 Discussion

Table 3 summarizes the general comparison made in Section 4 between the guidelines from both approaches. A bullet denotes that the identified category

Table 3. Consistency of NSBP guidelines and EO

Guideline	Consistent	EO-ignorant	Conflict
1.1 Elementary Business Process	●		
1.2 Elementary Life Cycle Information Object		●	
1.3 Aggregated Business Process			●
1.4 Aggregation Level		●	
1.5 Value Chain Phase			●
1.6 Attribute Update Request	○		○
1.7 Actor Business Process Responsibility	●		
1.8 Notifying Stakeholders		●	
1.9 Payment	●		
2.1 Product Type		○	○
2.2 Stakeholder Type		●	
2.3 Access Channel	○	○	
3.1 A Single Functional Task - Overview		●	
3.2 CRUD Task		●	
3.3 Manual Task		●	
3.4 Managing Time Constraint		●	
3.5 Business Rule Task		●	
3.6 Bridge Task	●		
3.7 Synchronization Task	●		
3.8 Synchronizing Task	●		
3.9 Actor Task Responsibility	●		
4.1 Unique State Labeling	●		
4.2 Unique State Property	●		
4.3 Explicit Business Process End Point	●		
4.4 Single Routing Logic	○	○	

(i.e., consistent, EO-ignorant or conflicting) is determined without any doubt. An open circle means the categorization still needs further elicitation as a unique categorization could not be identified.

When scanning the table, it can be argued that the theories comply on many points (i.e., at least 10 out of 25 guidelines are consistent), indicating that a surprising overlap exists between guidelines prescribed by EO and NSBP, given their different theoretical backgrounds. From a scientific viewpoint, this should encourage further initiatives to identify a potential common scientific basis. The EO-ignorant category is mostly discovered in the NSBP task rules. Almost all observations can be contributed due to the different abstraction level (EO does not consider these design questions), or the lack of a clear available answer in the different publications (e.g., Stakeholder Type). Consequently, NSBP seems to answer some design questions EO does not answer or does not consider. However, it should be stressed that the identification of both EO-ignorant (or potential NSBP-ignorant) guidelines does not indicate the superiority of one theory over the other. Rather, it might indicate certain areas which are less prominent in one theory when compared to the other (e.g., due to their different goals for engaging in modeling efforts). Regarding the conflicting guidelines, some genuine contradictions (e.g., Aggregated Business Process) were identified. These conflicts should be clarified in future research, especially because most conflicts occur in the core (i.e., the first twelve) NSBP guidelines.

When analyzing the case results, several remarks can be made as well. First, it can be noticed that most of the considered business processes were identified by both approaches. For instance, the EO transactions “membership registration” and "membership fee payment" easily maps on the NSBP "Create New Membership" and "Collect Yearly Fee" processes. This finding is consistent with the conclusion from our general comparison that guidelines resulting from both theories comply on most points.

Second, we also identified a set of differences in the case. The first difference noted in the case (i.e., regarding the Loan Process) could be categorized as a NSBP-ignorant situation. Indeed, whereas NSBP does not see any reason to to split this concern into multiple processes, EO does identify three processes (i.e., loan start, book return and loan) due to different created ontological facts. This cannot be directly derived from our general comparison made in Section 4, as our starting point was the set of guidelines from NSBP and investigate their consistency, ignorance or conflictation regarding EO. The second difference (i.e., regarding the Invoice Process) could be categorized as a possible contradiction between EO and NSBP. Whereas NSBP identifies an Invoice Lifecycle Information Object that requires its own business process, EO considers it as a request fact of the payment transaction. This aligns with the fact that one of the guidelines NSBP uses to identify this business process (i.e., Value Chain Phase) was equally labeled as contradicting previously. The third and fourth difference considered processes (i.e., regarding the Notification and Member Interaction Process) were only identified by the NSBP approach and not by EO. The authors do not consider these three differences in this case study to be contradicting but rather EO-ignorant as the two approaches inherently have another abstraction level on which they operate (cf. supra). This aligns with the fact that the Aggregation Level guideline and Notifying Stakeholders guidelines (which were used to identify the Member Interaction Process and Notification Process respectively) were equally labeled as EO-ignorant previously.

The difference between the identification of the loan process in NSBP and multiple transactions in EO illustrates the need for additional in-depth research. Currently, the primary constructs of NSBP and EO are compared: processes and tasks denote the modular structure in NSBP, and ontological acts and transactions denote the modular structure in EO. However, EO distinguishes an additional aggregation as well, being a group of transactions. These groups (in a publication also referred to as fibers) can be seen in a.o. the process model, which represents the interaction between transactions. Moreover, the state model in EO [7, p. 202] clarifies that the same Loan object is used for defining the production facts of all three transactions. This indicates that this Loan object in the EO state model resembles the Life Cycle Information Object of the Loan NSBP. However, since we compare NSBP processes to transactions, we actually compare the LCIO to the production facts, which are more specific. In a way, this observation is similar to the specification of multiple NSBP processes on a single LCIO, as suggested in NSBP [6, p. 142]. It should be clarified whether identified differences between NSBP and EO are actual differences, or are observed because different aggregation levels are compared.

Finally, although most business processes were identified by both approaches, the reasons behind their existence might differ. For example, the transaction

T03: Reduced fee approval is a separate transaction in EO because it is executed by a different actor [7, p. 144]. In NSBP, it is a different transaction as "it denotes a separate concern, because it recurs in at least two situations" [6, p. 217] (i.e., when creating a new member and when collecting the yearly fee).

Regarding future research, Section 3 and the first observed difference from the case study illustrated the need to research the NSBP-ignorant category as well. For instance, various coordination acts are not required to be modeled in NSBP, for example, when they are implicitly executed in business processes. The explicitation of this category in future research efforts could especially aid the completeness of NSBP models. Also, a more in-depth analysis regarding the different reasons for consistent design decision between both approaches is required, as was illustrated by the the library case. Nevertheless, the authors hypothesize that —given the consistency between both theories and under the condition that the different abstraction levels on which they clearly operate do outweigh the contradictions, or that contradictions could be resolved by clearly identifying the abstraction levels on which both theories have their proven scientific importance— a method combining both theories to analyze businesses can be proposed. Eventually, such method should contribute to more scientifically based and consistent mechanisms to build information systems, business processes and enterprise architectures in the future. In addition, it should be mentioned that by specifying NSBP, no direct mapping to NS software can be made yet, although first insights do seem promising.

7 Conclusion

In this paper, we explored to which extend the prescriptive guidelines related to the business process domain of EO and NSBP are consistent, complementary, or conflicting. We explained how both approaches offer theory-based guidelines to design business processes, and discussed in detail the assessment of the various NSBP guidelines. Moreover, we suggested several possibilities for further research, to work towards an integrated methodology for business process design.

Acknowledgments. P.D.B. is supported by a Research Grant of the Agency for Innovation by Science and Technology in Flanders (IWT).

References

1. Galbraith, J.R.: Organization design: An information processing view. Interfaces 4(3), 28–36 (1974)
2. Avenier, M.J.: Shaping a constructivist view of organizational design science. Organization Studies 31(9-10), 1229–1255 (2010)
3. Mendling, J., Reijers, H.A., van der Aalst, W.M.P.: Seven process modeling guidelines (7pmg). Inf. Softw. Technol. 52(2), 127–136 (2010)

4. Dietz, J.L.G., Hoogervorst, J., Albani, A., Aveiro, D., Babkin, E., Barjis, J., Caetano, A., Huysmans, P., Iijima, J., Van Kervel, S., Mulder, H., Op 't Land, M., Proper, H.A., Sanz, J., Terlouw, L., Tribolet, J., Verelst, J., Winter, R.: The discipline of enterprise engineering. International Journal of Organisational Design and Engineering 3(1), 86–114 (2013)
5. Mannaert, H., Verelst, J.: Normalized Systems—Re-creating Information Technology Based on Laws for Software Evolvability. Koppa, Kermt, Belgium (2009)
6. Van Nuffel, D.: Towards Designing Modular and Evolvable Business Processes. PhD thesis, University of Antwerp (2011)
7. Dietz, J.L.G.: Enterprise Ontology: Theory and Methodology. Springer (May 2006)
8. Mannaert, H., Verelst, J., Ven, K.: The transformation of requirements into software primitives: Studying evolvability based on systems theoretic stability. Science of Computer Programming 76(12), 1210–1222 (2011)
9. Mannaert, H., Verelst, J., Ven, K.: Towards evolvable software architectures based on systems theoretic stability. Software: Practice and Experience 42(1), 89–116 (2011)
10. McIlroy, M.: Mass produced software components. In: Naur, P., Randell, B. (eds.) NATO Conference on Software Engineering, Garmish, Germany, Scientific Affairs Division, NATO, Brussels, October 7–1, pp. 138–155 (1968)
11. Parnas, D.: On the criteria to be used in decomposing systems into modules. Communications of the ACM 15(12), 1053–1058 (1972)
12. Lehman, M.: Programs, life cycles, and laws of software evolution. Proceedings of the IEEE 68(9), 1060–1076 (1980)
13. Jackson, L.: Digital Filters and Signal Processing, 2nd edn. Kluwer Academic Publishers, Norwell (1988)
14. Oppenheim, A., Willsky, A., Nawab, S.: Signals & Systems, 2nd edn. Prentice-Hall, Upper Saddle River (1996)
15. Huysmans, P.: On the Feasibility of Normalized Enterprises: Applying Normalized Systems Theory to the High-Level Design of Enterprises. PhD thesis, University of Antwerp (2011)
16. Goldkuhl, G., Lyytinen, K.: A language action view of information systems. In: Ross, C., Ginzberg, M. (eds.) Proceedings of the 3rd International Conference on Information Systems (ICIS 1982), Michigan, pp. 13–31 (1982)
17. Searle, J.R.: Speech Acts: An Essay in the Philosophy of Language. Cambridge University Press, Cambridge (1969)
18. Bunge, M.: Treatise on Basic Philosophy: vol. 4: Ontology II: A World of Systems. Reidel, Boston (1979)
19. Flores, F., Ludlow, J.: Doing and speaking in the office. In: Fick, G., Sprague, R.H. (eds.) Decision Support Systems: Issues and Challenges, pp. 95–118. Pergamon Press, New York (1980)
20. Dietz, J.L.G.: The atoms, molecules and fibers of organizations. Data Knowl. Eng. 47, 301–325 (2003)
21. Winograd, T., Flores, F.: Understanding Computers and Cognition: A New Foundation for Design. New edn. Addison Wesley (1986)
22. Dietz, J.L.G., Albani, A.: Basic notions regarding business processes and supporting information systems. Requir. Eng. 10, 175–183 (2005)
23. Van Nuffel, D., Mulder, H., Van Kervel, S.: Enhancing the formal foundations of bpmn by enterprise ontology. In: Albani, A., Barjis, J., Dietz, J.L.G. (eds.) CIAO! / EOMAS 2009. LNBIP, vol. 34, pp. 115–129. Springer, Heidelberg (2009)

24. van Reijswoud, V.: Model based business system transformation. In: Pries-Heje, J., Ciborra, C., Kautz, K., Valor, J., Christiaanse, E., Avison, D. (eds.) Proceedings of ECIS 1999, Copenhagen, pp. 600–615 (1999)
25. Campagnolo, C., Camuffo, A.: The concept of modularity in management studies: A literature review. International Journal of Management Reviews 12(3), 259–283 (2010)
26. Sanchez, R., Mahoney, J.: Modularity, flexibility, and knowledge management in product and organization design. Strategic Management Journal 17, 63–76 (1996)
27. Reijers, H., Mendling, J.: Modularity in process models: Review and effects. In: Dumas, M., Reichert, M., Shan, M.-C. (eds.) BPM 2008. LNCS, vol. 5240, pp. 20–35. Springer, Heidelberg (2008)
28. Hevner, A.R., March, S.T., Park, J., Ram, S.: Design science in information systems research. MIS Quarterly 28(1), 75–105 (2004)
29. Peffers, K., Tuunanen, T., Rothenberger, M., Chatterjee, S.: A design science research methodology for information systems research. Journal of Management Information Systems 24(3), 45–77 (2007)
30. Winter, R., Albani, A.: Restructuring the design science research knowledge base. In: Baskerville, R., De Marco, M., Spagnoletti, P. (eds.) Designing Organizational Systems. Lecture Notes in Information Systems and Organisation, vol. 1, pp. 63–81. Springer, Heidelberg (2013)
31. Huysmans, P., Oorts, G., De Bruyn, P., Mannaert, H., Verelst, J.: Positioning the normalized systems theory in a design theory framework. In: Shishkov, B. (ed.) BMSD 2012. LNBIP, vol. 142, pp. 43–63. Springer, Heidelberg (2013)
32. Habermas, J.: The Theory of Communicative Action: Reason and Rationalization of Society, vol. 1. Beacon Press, Boston (1984)
33. Huysmans, P., Bellens, D., van Nuffel, D., Ven, K.: Aligning the constructs of enterprise ontology and normalized systems. In: Albani, A., Dietz, J.L.G. (eds.) CIAO! 2010. LNBIP, vol. 49, pp. 1–15. Springer, Heidelberg (2010)
34. Krouwel, M.R., Op 't Land, M.: Combining demo and normalized systems for developing agile enterprise information systems. In: Albani, A., Dietz, J.L.G., Verelst, J. (eds.) EEWC 2011. LNBIP, vol. 79, pp. 31–45. Springer, Heidelberg (2011)
35. Op 't Land, M., Krouwel, M.R., van Dipten, E., Verelst, J.: Exploring normalized systems potential for dutch mod's agility. In: Harmsen, F., Grahlmann, K., Proper, E. (eds.) PRET 2011. LNBIP, vol. 89, pp. 110–121. Springer, Heidelberg (2011)
36. Dietz, J.L.G.: Deriving use cases from business process models. In: Song, I.-Y., Liddle, S.W., Ling, T.-W., Scheuermann, P. (eds.) ER 2003. LNCS, vol. 2813, pp. 131–143. Springer, Heidelberg (2003)

Transformation of Imperative Workflows to Declarative Business Rules

Lex Wedemeijer

Department of Computer Science, Open University The Netherlands, Valkenburgerweg 177, Heerlen, Netherlands
Lex.Wedemeijer@OU.nl

Abstract. Business rules and workflow models are both advocated as a means to specify the way of working in the business, so overlap between the two may be expected. Business rules lay down guidelines and restrictions about the way of working in an organization. Workflow models specify which activities should be conducted in what order, when a trigger arrives. Thus, the constructs of the workflow models encapsulate the rules of processing. In this paper, we show how the main procedural constructs of imperative workflows can be transformed to declarative business rules. The transformation results in two rules that reflect the procedural nature of workflow. These capture the business requirements of work processing on a more abstract level with less emphasis on implementational detail than the corresponding workflow model. By transforming the workflows to declarative rules, the rules of the business become available for analysis, allowing the organization to extend their ruleset or to prune it for redundant rules.

Keywords: Business Rules, Requirements Engineering, Workflow Model, Declarative Model, Relation Algebra.

1 Introduction

Are you familiar with Sudoku puzzles? The rules are surprisingly simple, yet the challenge of Sudoku is that there is no simple workflow that will solve the puzzle. Reasoning may be done in any order of processing step, with the restriction that you comply with the rules at all times. We believe that something similar applies to workflows in large business administrations. The rules governing the day-to-day work may be rather simple, but workflows dictate a set of processing rules that often have little business relevance but that business workers and applications still must follow.

Business rules and workflow models are both advocated as a means to specify the way of working in the business. However, there is considerable overlap between these two. Business rules lay down guidelines and restrictions about the way of working in an organization. Workflow models specify, when an appropriate trigger is entered, which activities should be conducted and in what order, so that the users know what to do and when. We outline how these processing instructions can be extracted from the models and transformed into a single declarative business rule.

B. Shishkov (Ed.): BMSD 2013, LNBIP 173, pp. 106–127, 2014.

In keeping with the Business Rules Manifesto [1], we believe that business rules should be expressed as explicit constraints on behaviour, independent of how the rules may currently be implemented in process descriptions or workflow diagrams. It is generally claimed that the transformation of a workflow to declarative rules is possible, but it is rarely explained how to go about the transformation.

Employing the well-established formalism of Relation Algebra [2], [3], we prove this claim to be correct. The main contribution of this paper, found at the end of section 5, is the Imperative-Workflow Rule that exactly captures the imperative behaviour of an workflow. It employs a number of binary relations that correspond to the overall structure of the workflow model.

The importance of the transformation is to expose the processing rules, previously encapsulated in the imperative workflow model, at the same abstraction level and in a compatible format to other business rules. This enables an organization to manage the set of business rules as one coherent body [4]. The processing rules become amenable for practicable validation by the user community, and rule designers can conduct conflict analysis, elimination of duplicate rules, and overall optimization.

The paper is outlined as follows.

Section 2 discusses characteristics of workflow models, and our approach for the transformation of workflows into rules. Section 3 outlines declarative business rules. Next, we explain step by step the transformations of each of the four basic constructs of workflow models into partial rules. For the readers' convenience, the entire transformation is described in two parts. Section 4 outlines how we transform simple workflows without loops. In section 5 we explain how to transform the iteration loops as seen in more complex workflows. The result of the transformation is presented as a single Imperative-Workflow Rule at the end of this section. Section 6 introduces the End-to-End Rule that makes sure that every workflow process will be driven to its conclusion. Section 7 summarizes our findings. Section 8 concludes the paper.

2 Imperative Workflow

This section outlines main characteristics of conventional workflow models that we will be dealing with in the paper. We outline our approach for the transformation of imperative workflows into rules.

2.1 The Example Workflow

In 1997, the Workflow Management Coalition presented a definition of workflow as: 'a formalised view of a business process, represented as a co-ordinated (parallel and/or serial) set of process activities that are connected in order to achieve a common goal' [5]. A later report sets out the four basic constructs of typical workflows: sequence, parallel flow, selective flow, and iterative loop [6], [7]. In sections 4 and 5, we describe the transformation of these basic constructs into declarative rules.

Based on the WMC'99 technical report, a fictitious example of a workflow model containing all four constructs is shown in figure 1. The example, that we will be using throughout the paper, depicts the workflow of an editor receiving scientific papers to be reviewed for publication.

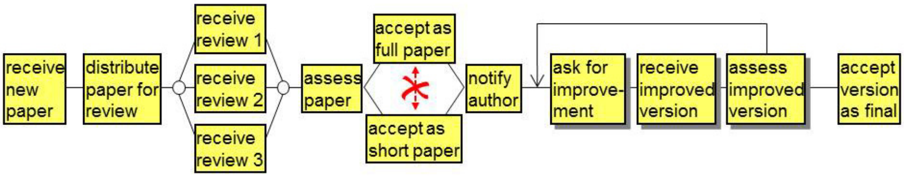


Fig. 1. Example workflow model

2.2 Imperative View of Workflow

Workflow models such as the one shown in figure 1 can be viewed in two ways.

One view of the flow is as a kind of roadmap. Once the workflow case is triggered, the next activities are executed in parallel or in series, until all work is done and, by assumption, the intended business goal of the workflow process is achieved. Whenever an activity is completed, the roadmap is consulted to answer the 'now what, where to go next' type of question. The answer is what may be called a 'statement of advice' [8]. Upon completion of activities, the roadmap indicates which activities in the processing chain may be executed next. Typical wordings of this kind are 'you may now start the activity named K', or 'activity K is now enabled for the case at hand'. This interpretation of workflow is forward-looking in time: what may come next, and we will refer to it as the *indicative* view of workflow

Our approach takes the other, more rigorous view: a workflow model specifies compulsory precedence. Prior to completing an activity, all of the preceding activities must also have completed, out of necessity. The question here is 'what must have come before', and the answer takes the form of a business rule. The rule is strictly enforced and may not be violated at any time. This looks backwards in time: what must have come before. We will refer to this as the *imperative* (view of) workflow.

2.3 Event-Based Approaches Towards Workflows

Many workflow modelling approaches are event-based, and use the 'incoming event' or 'trigger' as a core notion. The trigger starts the processing of a 'case' or 'workflow instance'. This constitutes an essential difference between workflow models and business rules in general: declarative approaches specify the business rules that should be complied with but do not refer to any particular 'case' being managed. If any rule is violated, there is work to do, regardless how or what caused the violation.

Many rule-based process modelling languages are based on the ECA (Event-Condition-Action) paradigm [9], [10]. The event component signals that some transition from one relevant status to another has occurred, and the condition specifies which action needs to be executed to deal with the event. The Petrinet paradigm particularly is a widely known transition-based approach which allows to study the properties of a workflow and its activities in great detail. An activity life cycle is described [11] that comprises steps like 'enable', 'allocate to resource', 'initiate work', transact data', 'record output', and 'finish', and the timestamp of each step may be recorded. Other interesting details of activities are resource responsible for enactment, execution cost, etc. Overall quality aspects of workflows can also be investigated, including features such as liveness and deadlocks in the model [12].

Our interest however lies only with overall workflow layouts such as figure 1, and whether an activity has completed or not. No timestamps are recorded: not the time of initiation, and neither the time of completion, nor are we interested in the duration of the activity. Overall quality issues of workflows are also beyond our scope.

Still, we find that a comparable notion of 'workflow case' is essential in our transformations of workflow models. However, our notion will have a slightly different meaning, which is why we use the abstract label 'identifier' in our model. The notion of workflow case and this identifier notion will be closely linked, but they are not quite the same.

Fahland et al. [13] do not distinguish between state- and event-oriented views of business rules. Moreover, their understanding of 'declarative' versus 'imperative' is somewhat differently. They define that 'in a declarative model, all requirements must be satisfied by the given behavior'. According to this understanding, a procedural workflow model that forces an organization to always adhere to the processing requirements should be dubbed 'declarative'. We do not concur with this. Rather, we define 'declarative' to mean that rules must be state-oriented, and there is no dependence on events, transitions, or sequencing.

2.4 Our Approach

The Business Rules Manifesto lays down a declarative approach to workflow. One of its articles states that 'rules are explicit constraints on behavior' and 'rules are not process and not procedure. They should not be contained in either of these'. This encourages to capture the workflow model in a non-procedural format, and business rules should be expressed as explicit constraints on behaviour, regardless of their current implementations in process descriptions or workflow diagrams.

In keeping with this manifesto, we aim to capture a workflow model as declarative business rules. We employ the imperative view of workflows, i.e. we will capture the rules on what must have 'come before' as a workflow is being executed. The indicative view about activities that may be executed next, is not studied.

In our approach, important information needed about a running case is what activities have completed or not. This calls for a notion of 'workflow case' or 'working instance', for which we will use the abstract label 'identifier', as illustrated in figure 3. But no other data about the execution of activities is used, such as initiation or completion time, resource allocation. The notion of time is irrelevant: our approach is state-oriented, and we do without a notion of transitions or events.

We focus on the overall structure of the workflow as a coordinated and connected set of process activities, and quality properties of workflow model are beyond our scope. We investigate the four main constructs of workflows in turn, and derive a partial business rule for each. By combining partial rules, we arrive at the sought-after business rule that captures a workflow model as a Relation Algebra expression.

3 Declarative Business Rules

This section introduces binary Relation Algebra as will be used in the paper.

3.1 Relation Algebra as Formalization Framework

We use binary Relation Algebra to specify and formulate declarative business rules, in the manner as described in Michels et al. [14]. To illustrate our application of the mathematical theory, a small excerpt from our running example is helpful.

Figure 2 depicts a few instances and their relations. By tracing lines in the diagram, the reader may verify the rule that if an activity is completed for some paper, then the preceding activity should also have completed. Thus, the paper "Modelling of KPI's" has *completed* the activity 'distribute paper for review', and so this paper must also have *completed* the preceding activity 'receive new paper'. Likewise, completion by the paper "Conflictation of Business Ontology" of the activity 'receive review 1' implies that its preceding activity 'distribute paper for review' must be *completed* by that paper. The example contains some violations of the rule, but our purpose is to just provide a small illustration. The proper version of the rule is presented in section 4.

Table 1. Example rule for scientific papers.

in natural language	in binary Relation Algebra
the preceding activity must be completed	[scientific paper] *completed* [activity type] ⊂ [scientific paper] *completed* [activity type] o [activity type] *precedes* [activity type]

The symbol o in this formula represents composition, also known as natural join, which takes two relations as its arguments, to yield a new binary relation.

The core element of this assertion is the set-inclusion or implication, denoted by the ⊂ symbol: if a tuple is present in the left-hand set, then that tuple must be present in the right-hand set.

There are some important differences with relational database modelling. Our 'concept' notion is comparable to entities, but it has only a single column. Thus, every instance of a concept is just a name which is key, and it comes with no attributes. Binary relations are not foreign-key pointers, but are defined as subsets of the Cartesian Product. Time is not a native notion of Relation Algebra. Indeed, none of the formulas and rules to be discussed in the paper will refer to time.

3.2 Alternative Formalizations

A wide range of modeling languages is available for modelling and runtime support of business processes and business rules. A comparative analysis of languages may be found in [15].

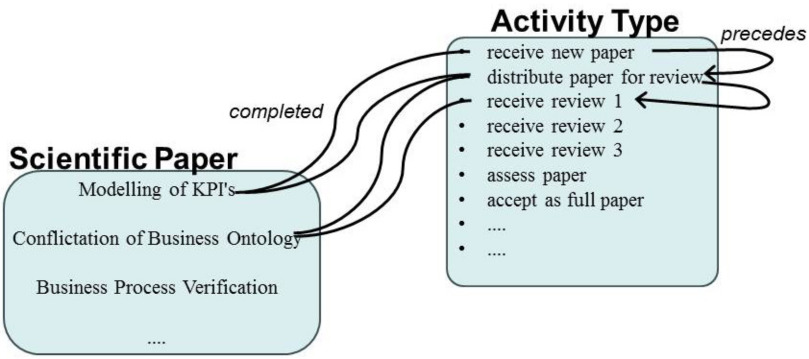


Fig. 2. Instance diagram for the example rule for scientific papers

As indicated above, we rely on Relation Algebra which is a variant of First-Order Logic. It enables to specify business rules with enough precision to implement directly into computer systems without additional translation steps.

Other variants of First-Order Logic may be used to formalize the rules encapsulated in workflow models.

We prefer Relation Algebra over less formal frameworks such as SBVR or RuleSpeak [16], [17], or Controlled Natural Languages [18]. These formalizations lack computer precision, and the rules expressed in these languages require additional translations in order to implement them in computer systems.

Linear Temporal Logic (LTL) extends First-Order Logic with a linear, discrete notion of time which makes for an excellent basis to study workflows in great detail [19]. A prototype workflow system, called Declare [20], [21] uses LTL as its language to describe business process models and to manage run-time process execution. A main difference with our approach is the notion of time, a prime feature in the LTL approach but absent from ours.

Protocol modelling [22] is an event-based approach that does away with temporal aspects and the notion of cases. The approach employs aspect-oriented models that enable state-transitions while taking multiple cross-cutting concerns and business constraints into account. This approach takes the indicative view of workflow when it labels state transitions as 'desired' [23].

3.3 Basic Structure in Relation Algebra

To transform imperative workflows into the declarative format in Relation Algebra, we must first specify a suitable structure. The basic structure of our declarative model (figure 3) is rather simple, with just two concepts and one relation. We will expand this structure with other relations later.

The [identifier] concept represents (a pool of) available case identifiers. Each identifier is associated with a single workflow case, e.g. the paper named "Conflictation of Business Ontology". We will elaborate on this in section 5.

The [activity type] concept represents (the set of) activity types. When executing a workflow case, each activity type may be instantiated (executed) zero, one or perhaps several times, in order to achieve the goal of the business process. The word 'activity' may be used for actual executions of an activity type, but in this paper we have little use for this word as we rarely need to refer to such instantiations.

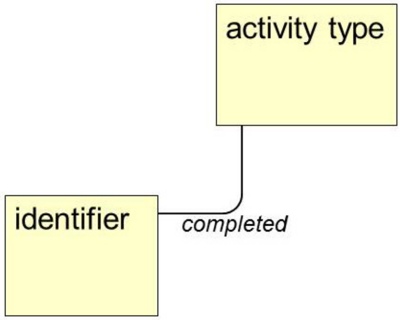


Fig. 3. The declarative model

The most important relation of all, *completed*, records workflow progress (see table 1). A tuple (i,A) in this relation *completed* means that this case identifier, i, has been successfully processed by the particular activity type, AT.

Table 2. The relation *completed*

relation	semantics
[identifier] *completed* [activity type]	is: (the recording that) all work of the activity type has been successfully completed for (the workflow case associated with) identifier i.

The completed relation represents the audit trail of the work done on a particular case. In accordance with compliance regulations and good records-keeping [24], tuples may be added into this relation, but they may never be altered or deleted thereafter: an activity cannot be un-completed. And to safeguard referential integrity, we cannot delete an identifier or activity type once it is recorded in the *completed* relation. Notice that we abstract from a lot of attributes commonly included in audit trails, such at deadlines being set, the exact times of start and completion, business resource that executed the work, or the actor taking responsibility for the work done.

We will assume the completed relation to be total, i.e. a case identifier shall be recorded only if it completed at least one activity. We are only interested in identifiers associated with actual work done, not in possible future work. The reverse is not required: an activity type may exist even if no case has ever completed that activity.

4 Transformation of Forward Workflow

This section outlines how three common components of workflows, sequence, synchronize (AND-join), and disable (OR-split), can be transformed into a declarative rule. The three partial rules are merged into a single Forward-Workflow Rule.

4.1 Sequence / Precedence

Sequence / Precedence in Workflows. Precedence is the commonest construct of workflows. Most workflow diagrams depict such sequencing by an arc pointing from one unit, representing an activity type A_M, to a next one labeled A_N**.**

The indicative interpretation of such sequencing is: if completion of A_M is on record, then the rule says that A_N can be recorded as *completed* next. Recording that A_M has *completed* but not A_N is allowed, for a while, and does not violate the imperative sequencing that the workflow dictates. In the imperative interpretation of workflow, an arc from A_M to A_N imposes strict precedence: whenever the next activity A_N is *completed* for some case identifier, then A_M must also have *completed* for that identifier. Or: if completion of A_M is not on record, then the rule says that completion of A_N is impossible. Having completed A_N but not A_M violates the imperative workflow rule, and this violation is never permitted.

Precedence in our understanding restricts only the completion of activities. In some interpretations however, precedence is taken to mean that an activity cannot be started before completion of the preceding one. Differentiating between start- and end times would imply that activities have a certain duration, but as noted before, time is irrelevant to our approach and we do not follow this interpretation.

Sequence / Precedence Transformed to Declarative Rule. To capture precedence as a declarative rule, the *precedes* relation on activity types is introduced (see table 3)

Table 3. The relation *precedes*

relation	semantics
[activity type] *precedes* [activity type]	is: the precedence relation among activity types. A tuple (A_M, A_N) in this relation means that only if an activity of type A_M has (been recorded as) completed, should a corresponding activity of type A_N also be (recorded as) completed.

The precedence rule is formulated in first-order logic using this *precedes* relation:

for each $i \in$ [identifier], and each $A_N \in$ [activity type], we have:
if i *completed* A_N then
for at least one activity type A_M it holds that A_M *precedes* A_N and
i *completed* A_M

which in Relation Algebra reads:

$$completed \subset completed \circ precedes \,. \qquad \textbf{(1)}$$

Sequence / Precedence for Triggering Events. A point with the rule above is that not every activity type is preceded by another. In particular, triggers are exceptional, and they are important because they set the workflow in motion. At first glance, starting activity types invalidate the assertion (1). Completion of a new case instance cannot be recorded for an initial activity type because a proper tuple in the *precedes* relation is absent. We solve this by adapting the relation *precedes*: initial activity types are recorded by way of self-referring tuples (A_0,A_0). By inserting such tuples into *precedes*, the assertion (1) also covers initial activities of the workflow. In the example workflow of figure 1, we need to insert a tuple ('receive new paper', 'receive new paper') to make this the unique starting point.

Sequence / Precedence Captured by the *Precedes* Relation. Basic properties of the binary *precedes* relation can be deduced from behavioural properties of imperative workflows. As we made sure that every activity type is preceded by at least one other, it is clear that the binary *precedes* relation is surjective. The *precedes* relation is not univalent, as an activity type may well precede several others, corresponding to a so-called split in the workflow. It establishes what may be called a 'multiple instance pattern' [25], and subsequent activities may be executed and completed in parallel along separate branches of the flow. Relation *precedes* is not total, an activity type may be a 'last one' or terminating activity in the workflow. Nor is *precedes* an injective relation, as more than one activity type may precede an activity type A_N, this is the common OR-join of workflow models. And for ordinary workflows, *precedes* is irreflexive, asymmetric and acyclic, except of course for the triggering activity types.

4.2 Synchronizing AND-Joins

AND-Join in Workflows. Sometimes an activity may only be completed after the completion of two or more activities that are processed in parallel. This behaviour is seen in the example workflow of figure 1 in activity type "assess paper" which should be completed only after having received all three reviews that are executed in parallel. In workflow models, this is known as an AND-join, or more formally a synchronization point. Our assertion (1) captures OR-join behaviour, not AND-joins.

AND-Join Transformed to Declarative Rule. To properly capture AND-joins in a declarative rule, we introduce relation *multi-precedes* for activity types (see table 4).

Table 4. The relation *multi-precedes*

relation	semantics
[activity type] *multi-precedes* [activity type]	is: an activity type precedes a next activity type, and its execution must be synchronized. A tuple (A_M,A_N) in this relation means that only if the activity of type A_M and certain others too have completed, may a corresponding activity of type A_N also be completed.

The rule for AND-join synchronization can now be formulated in first-order logic:

for each i∈ [identifier], and each A_N∈ [activity type], we have:
if i *completed* A_N then
for all activity types A_M such that A_M *multi-precedes* A_N, it holds that
i *completed* A_M

This is formulated in Relation Algebra by way of the left demonic composition operator [26], denoted here by the square bracket] symbol:

$$completed \subset completed \;] \; multi\text{-}precedes \; . \tag{2}$$

AND-Join Captured by the *Multi-precedes* Relation. The *multi-precedes* relation closely resembles the *precedes* relation described earlier. Indeed, the relations implement identical behaviour wherever there are just single precedents. They differ at join-points in the workflow with several activity types 'coming before'. Whereas *precedes* captures OR-join behaviour, *multi-precedes* models AND-join behaviour. As workflows can display both types of behaviour, both relations are needed. The *multi-precedes* relation has properties similar to the *precedes* relation: it is not univalent, nor total, nor surjective, and it is asymmetric and acyclic. It is not injective as AND-join behaviour assumes at least two, not at most one activity to be related to a subsequent one. Unlike *precedes*, this *multi-precedes* relation is irreflexive, as there is no synchronization of triggers.

4.3 Selective XOR-Splits and Disabling

XOR-Split in Workflows. Selective flow, or exclusive-or, also known as conditional branching, choice, means that some activity precedes two (or more) activities that are placed in parallel but only one of the succeeding activities is allowed to complete.

XOR-Split Transformed to Declarative Rule. In the workflow of figure 1, consider the OR-split after activity type 'assess paper' which *precedes* both the 'accept as full paper' and 'accept as short paper'. Evidently, only one of them is allowed to complete: completion of one prohibits completion of the other one. We cover this restriction by a new relation on activity types called *disables* (see table 5).

Table 5. The relation *disables*

relation	semantics
[activity type] *disables* [activity type]	is: the disabling relation among activity types. A tuple (A_X, A_Y) in this relation means that never if an activity of type A_X has completed, may a corresponding activity of type A_Y also be completed.

This rule for selective flow can thus be formulated in first-order logic as:

for each i∈ [identifier], and any pair A_X, A_Y∈ [activity type], we have:
if i *completed* A_Y then
not i *completed* A_X for any activity type A_X that *disables* A_Y

Denoted as a Relation Algebra assertion this reads:

$$completed \subset \neg\ (\ completed \circ disables\ )\ . \qquad \textbf{(3)}$$

XOR-Split Captured by the *Disables* Relation. Most activities are not involved in disablings, and therefore the disables relation is neither total nor surjective. One activity type may disable, or be disabled by several others, hence the relation is neither univalent nor injective. Evidently, the homogeneous *disables* relation is irreflexive, while nothing can be said about its being transitive or not.

The *disables* relation is used to capture XOR-splits. When used for this purpose, it is symmetric: the XOR-split means that if one activity type A_X completes, then the other activity A_Y may not complete, and reversely: if an activity of type A_Y is recorded as completed, then no corresponding activity of type A_X may be completed.

Interestingly, the *disables* relation may also be used to implement unilateral disables, which is a more advanced feature of some workflow models. Used in this way, the *disables* relation would no longer be symmetric.

Beware however of a potential problem in the timing of activity completion. Some workflow models interpret an activity type A_Y to be *disabling-for* activity type A_Z to mean that completion of A_Z is not allowed later than A_Y, but it is allowed prior to A_Y. This would allow to record first a tuple (i,A_Z) as completed, to be followed by (i,A_Y), but it would prohibit a tuple (i,A_Y) first and (i,A_Z) later. The sequencing of these recordings depends on actual timestamps, that we explicitly abstracted from in our analysis. Hence, our rules cannot deal with time-dependent disabling.

4.4 Forward-Workflow Rule

The three basic patterns of workflow analyzed so far all ensure a forward flow, in contrast to the flow that we will be analyzing in the next section. So far, we added three homogeneous relations to the basic model, resulting in the declarative model depicted in figure 4. The Relation Algebra assertions (1), (2) and (3) acquired for each of the three forward flow constructs, are easily combined into the single assertion (4).

RULE Forward-Workflow AS

$$\begin{aligned} &completed \subset \\ &\quad (\ (\ completed \circ precedes\ ) \cup (\ completed\ ]\ multi\text{-}precedes\ )\ ) \\ &\quad /\ (\ completed \circ disables\ )\ . \end{aligned} \qquad \textbf{(4)}$$

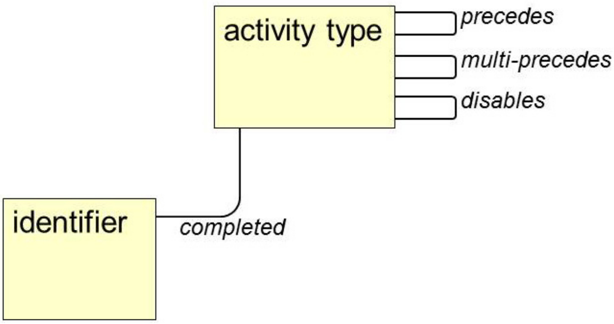


Fig. 4. Declarative model, extended for Forward Workflow

For later reference, the right-hand side of the assertion (4), after the ⊂ inclusion symbol, is called the *forward-flow* relation (see table 6). It is a derived relation and as such, it cannot be edited, unlike the base relations defined earlier.

Table 6. The derived relation *forward-flow*

relation	semantics
[identifier] *forward-flow* [activity type]	is: the relation with a tuple (i,A) indicating that for the identifier i, at least one (regular) precedent of A has completed, or all of its multi-precedents have completed, while none of its disabling activities have completed.

The contents of the *forward-flow* relation is reassessed every time that some new tuple is recorded in the *completed* relation. As a result, one or several new tuples may emerge in the *forward-flow* relation. Or, a tuple that previously was in the *forward-flow* relation, may disappear from the *forward-flow* because of disabling or some other feature.

The Forward-Workflow Rule is enforced rigorously: completion is always prohibited if the tuple is absent from the *forward-flow* relation. But, as the naming suggests, the rule holds for forward flows only, and does not apply for 'loops' or 'backward' flows that we will analyze in the next section.

5 Transformation of Iterative Loops

This section deals with the fourth construct, iteration or looping.

5.1 Iterations in a Workflow

Handling a workflow case will often involve the repeated execution of activities, until some business condition is met. The Forward-Workflow Rule described above cannot deal with a flow that double back onto itself. This is because binary relations may record a tuple once, but not several times over. Hence, *completed*, as a binary relation, cannot record repetitive completion of the same activity type by the same identifier.

Our solution is to employ a new identifier for each iteration of the loop. In business environments, tools for audit trailing, activity logging and process mining use similar solutions [27]. As a consequence, a workflow case is no longer represented by a single identifier. Instead, the workflow case is associated with the entire collection of identifiers used in its loop iterations. By expanding the definition of the 'identifier' concept in this way, the analysis and results of the previous section remain valid, if we properly attend to a number of details.

5.2 Relations to Model the Iterations

We capture iterations by imagining the execution of the workflow-case to pause at the looping activity where it may 'fire' zero, one or more iterations, depicted in figure 5. In the figure, execution of identifier i can be thought of as being suspended, and only when all of its iterations have been dealt with, can the main workflow-case complete the looping activity, and proceed in the normal way. At the right-hand side, the loop is unfolded to depict two iterations, labeled j and k, normally not shown in graphical models of the workflow.

The idea of identifier i to be suspended however should not be taken literally, as our approach has abstracted from duration of an activity and we record its completion only. Instead, we will prohibit for identifier i that the activity is *completed* if iterations are still running. This is somewhat different from the standard understanding of iterations, where a loop starts not during execution, but only after completion of the looping activity. As far as we could assess, this detail has no consequences for validity of our transformation.

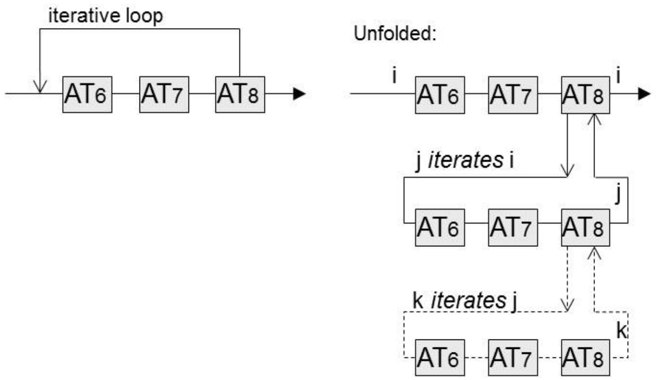


Fig. 5. Iterations use subordinate identifiers

Several new relations are needed to help us capture iterative loops. First, relation *loops-to* records the iteration loop between activity types, as drawn in the workflow diagram (see table 6). To keep things simple, we will assume that this relation on activity types is univalent (there is at most one outgoing loop for an activity type) and injective (at most one incoming loop for an activity type). Moreover, it is assumed that the target activity type (where it loops to) lies before the activity type where it originates. This is to ensure that a case looping back, will eventually return to the activity type that fired it.

To track which iterative loops are running for what workflow cases, we coin two more relations, *fired-from* and *iterates* (see table 7).

The *fired-from* relation is a function. It prevents possible confusion about which identifier originates where, in case a workflow model contains several loops. For the sake of consistency, activity types recorded in the *fired-from* relation must be present in the *loops-to* relation, but here again, we take this quality issue for granted.

The *iterates* relation for sub-identifiers is a function: an identifier *iterates* exactly one other. Remark that a sub may again fire its own sub-subordinates, and a stack of arbitrary depth may exist. Also remark that the *iterates* relation allows to fire several subordinates at once, for instance if a complex scheduling problem is broken down into several other scheduling problems, to be analyzed in parallel. Still, an identifier should not iterate itself or any of its subs, and the *iterates* relation is definitely irreflexive, asymmetric, and acyclic.

Table 7. The relations *loops-to*, *fired-from* and *iterates*

relation	semantics
[activity type] *loops-to* [activity type]	is: the iterative-loop relation of activity types. A tuple (A_N,A_K) in this relation means that an activity of type A_N, before it is *completed*, may invoke no, one or more iterations of the loop starting from A_K.
[identifier] *fired-from* [activity type]	is: the subordinate case identifier being started by the looping activity type. A tuple (j,A_M) in this relation means that identifier j is 'fired' from activity type A_M.
[identifier] *iterates* [identifier]	is: the iteration for identifiers. A tuple (j,i) in this relation means that the identifier j is 'fired', when it is decided that an iteration for the identifier i is required. An identifier that iterates another is referred to as the subordinate, or sub. The other identifier is called the main-case identifier.

5.3 Rules for Iteration

To transform iterations in imperative workflow into declarative business rules, we must account for several restrictions that apply to the identifier of the iteration:

- the restriction on completion of the initiating activity for this sub-identifier,
- the restriction on completion of its subsequent activities, up to and including the loop activity, and
- the restriction that it must terminate there, and complete no more activities.

And there is one restriction regarding the main-case identifier:

- we must make it 'wait for' all of its iterations.

First, remark that the activity type where an iteration begins, is not a triggering activity type in general. Hence, the Forward-Workflow Rule would normally prohibit the activity type from being *completed* by this particular identifier, as it lacks its proper precedent. The work-around of course is to employ a substitute identifier as its predecessor, viz. the main case that is firing this iteration.

Firing an iteration by a main-case identifier may occur only if that main-case identifier is enabled for the firing activity type, but has not yet *completed* that particular activity, i.e. it is in the *forward-flow* relation that we defined earlier.

Abstracting from the specific business knowledge that determines whether or not the iterative loop should be invoked and a subordinate case started, we can write down the imperative workflow condition to permit the initiating activity to be completed:

for any $j \in$ [sub], and any $A_K \in$ [activity type] where some A_M *loops-to* A_K:
if j *completed* A_K, and j *iterates* some identifier i,
then must the tuple (i, A_M) be in the *forward-flow* relation

Denoted as a Relation Algebra assertion this reads:

$$\textit{completed} \subset \textit{iterates} \circ \textit{forward-flow} \circ \textit{loops-to} \,. \qquad \textbf{(5)}$$

Formula (5) controls whether a new subordinate may complete its first activity.

When an iteration has *completed* its initial activity, it should go forward and complete the activity types that are part of this loop, up to and including the one where it was fired. This is normal processing as already described by the Forward-Workflow Rule, so no additional rules are needed.

But then we must ensure that the iteration terminates at its point of origin, where it was fired from. Going beyond that point and completing some activity further down the flow is prohibited. In particular, no activity type *completed* by a subordinate identifier, should lie beyond the activity type that fired the sub. For ease of use, we simplify this restriction somewhat and apply it only to the first activity type that lies beyond the firing activity type. In the example workflow of figure 1, this comes down to prohibiting the sub-identifiers j and k from completing the "accept version as final" activity. In first-order logic, the restriction reads:

for any $j \in$ [sub], and any $A_N \in$ [activity type], we have:
if j *completed* A_N, then it is never permitted that
j *fired-from* some activity type A_M that *precedes* A_N

Denoted as a Relation Algebra assertion this reads:

$$\textit{completed} \subset \neg\, (\textit{fired-from} \circ \textit{precedes}\,) \,. \qquad \textbf{(6)}$$

Finally, we need to make the main case 'wait at' the looping activity. It must wait there for its iterations, if any were fired, and may not proceed, or else a running iteration would become orphaned. The main case may complete only when iterations that were fired for it, have all run their course to completion. This is formulated in first-order logic:

for any i∈ [identifier], and any A_M∈ [activity type], we have:
if i *completed* A_M, then no sub exists that *iterates* this identifier i,
which was *fired-from* A_M and it has not *completed* activity type A_M

We can write this as a Relation Algebra assertion, using the ~ symbol to stand for inversion, applied here to the *iterates* relation.

$$completed \subset \neg\, (\, iterates{\sim} \circ (\, fired\text{-}from \cap \neg\, completed\,)\,)\,. \qquad \textbf{(7)}$$

The assertion is trivially satisfied if no iterations are fired. It is also satisfied if running iterations for a case do exist, but those were fired for another loop, from some other activity type in the workflow than the one about to be completed by the case. Also notice how assertion (7) applies recursively, i.e. it captures nesting of iterations. If a subordinate identifier fires sub-subordinates of its own, then it too will have to 'wait for' its own sub-subordinates before being allowed to complete.

5.4 Imperative Workflow Rule

Conditions (5) and (6) determine a scope for a subordinate identifier. They govern the start and termination of each subordinate. The two conditions, together with the Forward-Workflow Rule, adequately describe the workflow behaviour of subordinates without sub-subordinates of their own. The partial conditions can be combined into one Relation Algebra assertion (8):

RULE Subordinate-Workflow AS

$$\begin{aligned} &completed \subset \\ &\quad (\, forward\text{-}flow \cup iterates \circ forward\text{-}flow \circ loops\text{-}to\,) \\ &\quad /\, (\, fired\text{-}from \circ precedes\,)\,. \end{aligned} \qquad \textbf{(8)}$$

For main workflow cases, and for sub-identifiers that do fire sub-subordinates of their own, this rule (8) coincides with the regular Forward-Workflow Rule (4), except at activity types where iterative loops can be initiated. Therefore, we only need to merge condition (7) that controls behaviour at looping activities into the rule above:

RULE Imperative-Workflow AS

$$\begin{aligned} &completed \subset \\ &\quad \neg\, (\, iterates{\sim} \circ (\, fired\text{-}from \cap \neg\, completed\,)\,) \\ &\quad \cap\, (\, (\, forward\text{-}flow \cup iterates \circ forward\text{-}flow \circ loops\text{-}to\,) \\ &\qquad /\, (\, fired\text{-}from \circ precedes\,)\,)\,. \end{aligned} \qquad \textbf{(9)}$$

Our transformation of the imperative workflow produced this Imperative-Workflow Rule (9) which exactly captures the workflow behaviour in accordance to the imperative view. It prohibits an identifier to complete any activity in violation of the workflow.

As a bonus, the rule provides us with an indicative view. The rule can be read in lay terms as: 'if a tuple (i,A) is present in the right-hand side relation of (9), then tuple (i,A) may also be present in the left-hand side'. In other words, the right-hand side of the rule assertion indicates for identifier i which activity types either have *completed*, or are allowed to complete. It is fairly easy to derive from this an *is-enabled* relation, in the same way that we derived the *forward-flow* relation of section 4.

6 End-to-End Rule

The previous sections detailed how to capture the four basic constructs of workflows, leading up to the Imperative-Workflow rule. In this section, we introduce a new rule, called End-to-End rule to ensure that a workflow process, once it is triggered, will run to completion.

6.1 Workflow Completion

The definition of workflow presented by the Workflow Management Coalition stated that it aims 'to achieve a common goal'. We take this to mean that the workflow's goal is to complete its terminating activity. Thus, once the workflow is triggered, it ought to run its course up to its terminating activity. It is readily seen that this is not captured in the Imperative-Workflow Rule that prohibits an identifier to complete activities in violation of the flow. But that rule does not ensure that any activities are completed at all.

In the imperative view of workflow, the goal of the process is not achieved as a matter of course. If a workflow process halts in mid-term, nothing goes wrong, no rule is violated, no signal is raised that there is work to do. There is no guarantee that a case, once it started, shall be processed to completion. To ensure that the intended business goals of a workflow are met, we need to make sure that it will run from start to finish. Once a triggering activity is *completed* by a workflow case, it should always progress to its terminal activity. In a similar fashion, each subordinate iteration fired from some looping activity type ought to return to its point of origin.

To capture this, we formulate a new rule called the End-to-End Rule. And whereas the Imperative-Workflow Rule may never be violated, the End-to-End Rule permits violations, provided that all violations will be resolved in due course. Appropriate workflow activities should be executed and completed until no more violations emerge and the goal of the workflow is achieved.

6.2 About the *Progresses-to* Relation

To capture the End-to-End rule, we coin a new relation *progresses-to* from trigger to terminating activity type (table 8). To avoid trivial workflows, we require this relation to be asymmetric. As every trigger must lead to some goal, we require that every triggering activity type is related to at least one terminating activity type, i.e. when restricted to triggering activities, the relation is total. Likewise, every terminating activity type must be related to at least one trigger. On the other hand, it need not be univalent or injective.

Table 8. The relation *progresses-to*

relation	semantics
[activity type] *progresses-to* [activity type]	is: the relation that describes the overall end-to-end structure of the workflow. A tuple (A_S,A_T) in this relation means that A_S is a starting (initial) activity type, and A_T is a terminating activity type in the corresponding workflow.

The End-to-End Rule now states that every workflow trigger should always progress to its terminal activity or activities. The rule can be stated as:

for any $i \in$ [identifier], and $A_S \in$ [activity type], we have:
if i *completed* activity type A_S which *progresses-to* activity type A_T,
then must that identifier i *completed* that activity type A_T

The End-to-End Rule in Relation Algebra becomes:

$$completed \circ progresses\text{-}to \subset completed\ . \qquad \mathbf{(10)}$$

In contrast to previous rules, a temporary violation of this rule is permitted. For this rule, it is desirable to signal violations because it indicates that a workflow case is running and has not finished yet.

7 Discussion

We analyzed the four main constructs of common workflows. We defined two concepts and eight relations in a Conceptual Model based on binary Relation Algebra, as depicted in figure 6. We formulated two declarative business rules that capture the imperative workflow behaviour.

7.1 Results

We claim that our rule-based approach has enough expressive power to transform workflow models, and we formulated the declarative business rules as Relation-Algebra assertions.

One important reason for us to use binary Relation Algebra to express business rules is that its statements can be implemented directly into computer systems, no further translation is required. We implemented the declarative model, including the Imperative Workflow Rule and End-to-End Rule (see figure 6) in a tool environment that supports Relation Algebra, called Ampersand [28], and it was successfully verified, thus underpinning our claim.

A workflow constructed from the four constructs, sequence, AND-join, XOR-split, and iterative loops, may be described by populating the concepts and binary relations.

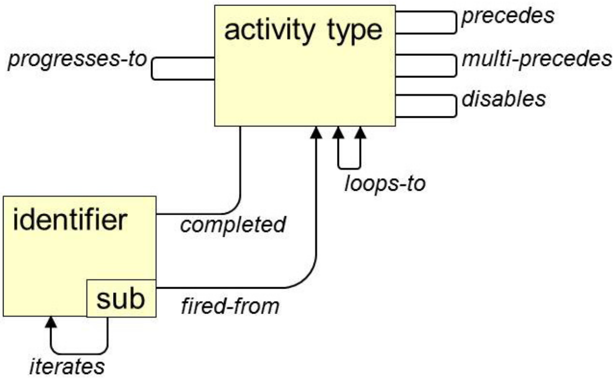


Fig. 6. Declarative model for Imperative Workflow and End-to-End Rule

The relations and rules of our approach can be characterized as follows:

- they capture all four constructs of common workflow models,
- they implement the imperative behaviour of the procedural workflow of the business process,
- they are declarative in nature, involving only (persistent) states and not volatile events or transitions,
- they are time-invariant, no notion of time is used, there is no 'before' and 'after'
- the two rules ensure that the process activities execute in a well-coordinated fashion. One rule makes it impossible to deviate from the imperative workflow model. The other rule forces to resolve all violations so that all terminating activities will be *completed* and the workflow goal is achieved.

The declarative model shown in figure 6 is not restricted to a single workflow, but has a more general validity. The same declarative model can be used to capture other workflow models as well. It suffices to populate the various binary relations with the appropriate data representing the constructs and their connections laid out in the workflow models. The behaviour of all such workflow models will then be correctly controlled by our two rules.

7.2 Limitations of Our Approach

As our approach abstracted from time, some usual features of workflows cannot be accounted for. For one, deadline expiry and the need for timely escalation procedures cannot be dealt with. This is because we cannot determine a starting time, or calculate an expected end time to detect the failure to complete. For another, a 'disable' that is specified with time-dependence could not be transformed into our declarative format.

In practice, other exceptions exist that operational workflows must deal with, such as lack of resources, user-initiated aborts, and crosscutting events. Likewise, quality problems may arise in workflows, such as deadlock, irregular termination, or loops that never terminate. Transforming to declarative rules cannot be expected to solve such quality problems, and this area of research is beyond the scope of this paper.

The Business Rules Manifesto defines a set of rules to be 'declarative' only if there is no implicit sequencing within that ruleset. As our transformation retains all procedural information about the sequencing, splits, joins and loops in the workflow, our ruleset would not qualify as 'declarative' rules. However, our understanding of 'declarative' is that there is no *explicit* dependence on events, transitions, or timing sequence in the rules, and our rules are state-oriented, not event-oriented.

7.3 Extensions

The sequencing of activities in a workflow is the outcome not only of business requirements, but also of design decisions and implementation choices. Another designer may come up with a different sequence that also complies with the essential business rules. Hence, precedence analysis is required to bring out what aspects of the flow is due to design choices, and which are based on actual business needs. To some extend, is a matter of opinion whether the workflow constitutes legitimate business rules, or whether it is just a way to implement underlying, more fundamental business rules [29].

Moreover, flow rules such as precedences, disablings and the like, are just one of the many types of business rules. Business rules in general support not only the consecutive steps of process flows but also the rules to assess business facts and classify events used within the executing activities for decision making. For instance, workflow diagrams often specify the decision rules that determine whether iteration is required, or which branch in an OR-split ought to execute. In our analysis of the workflow model, we have abstracted from such decision rules. Those decisions are usually based on particular information concerning the business case at hand. In our running example of the paper submission process, a rule for paper assessment might be "if two or three reviews rate the paper as above-average, then accept as full paper". Such a rule mixes content-aware data (the ratings of a paper) with flow control (which activity should or should not be completed next).

Future work is to augment our rules with content-aware rules, such as the criteria for iterations and OR-splits, and also the implicit decision rules that are currently encapsulated in the workflow activities. The Business Rules Manifesto advocates that there should be one cohesive body of rules, enforced consistently across all relevant areas of business activity, we envision a ruleset that is consistent and comprehensive, reflecting all the processing needs of the business.

Once such a ruleset is established, a next step is to check with business users how the workflow precedences and the like, now captured in binary relations and declarative rules, correspond to the requirements of their business environment. Also, it can be debated with users which flow features must be strictly enforced, and which ones may allow temporary violations. The aim is to support users with rule-based recommendations about which activities to execute in order to arrive at the goal of the business process [30].

8 Summary and Conclusions

In this paper, we prove the claim that imperative workflows can be captured as declarative business rules. Our proof is constructive: we show how to do this, by providing exact specifications of the declarative business rules.

The approach taken is to analyze the four main constructs of workflows, sequence, AND-join, XOR-split, and iterative loops. We outlined how to capture each of these workflow structures in eight binary relations involving just two concepts. One is the identifier concept, representing the workflow cases to be handled and subordinates when cases go through interactive loops in the workflow. The other concept is activity type workflows. The behaviour of the imperative workflow is then captured by two rules that involve these eight relations. For both rules, we established the exact formulas in binary Relation Algebra.

The first rule, called Imperative-Workflow Rule, captures the entire imperative workflow, allowing no violations at any time. This comprehensive rule was established in two steps. In the first step, we analyzed forward flow, which involves normal and parallel sequence, multiple precedence, and exclusions (selection). In the second more complicated step, we analyzed iterative loops.

The second rule is the End-to-End Rule, which drives the workflow through to its completion. This rule does allow violations, but while violations exist, there is work to do resolving them. Case handling is finished when there are no more violations, and the goal of the business process is reached.

By specifying exactly how to map the constructs of workflow to binary relations, we provided tangible evidence that the way of doing business may indeed be captured in a business rules model that meets all the demands of the Business Rules Manifesto.

We expect that declarative rules, developed along these ideas, will capture the business requirements about the processing of incoming work better than rigid rules of imperative workflows do. The ruleset will provide an essential basis for improved models to coordinate business processes.

References

1. Business Rules Manifesto: `http://www.businessrulesgroup.org/brmanifesto.htm`. Version 2.0. Edited R.G. Ross (last accessed December 14, 2013) (2003)
2. Tarski, A.: On the calculus of relations. Journal of Symbolic Logic 6(3), 73–89 (1941)
3. Maddux, R.: Relation Algebras. Studies in Logic and the Foundations of Mathematics, vol. 150. Elsevier Science (2006)
4. Kovacic, A.: Business renovation: Business rules (still) the missing link. Business Process Management Journal 10(2), 158–170 (2004)
5. Lawrence, P. (ed.): Workflow Management Coalition: Workflow Handbook 1997. John Wiley and Sons, New York (1997)
6. Workflow Management Coalition: Terminology & Glossary. Technical report WFMC-TC-1011 issue 3.0 (1999)
7. Lu, R., Sadiq, W.: A survey of comparative business process modeling approaches. In: Abramowicz, W. (ed.) BIS 2007. LNCS, vol. 4439, pp. 82–94. Springer, Heidelberg (2007)
8. Witt, G.: Writing Effective Business Rules. Morgan Kaufmann (2012)
9. Kappel, G., Rausch-Schott, S., Retschitzegger, W.: Coordination in Workflow Management Systems, A Rule-Based Approach. In: Conen, W., Neumann, G. (eds.) ASIAN-WS 1996. LNCS, vol. 1364, pp. 99–120. Springer, Heidelberg (1998)
10. Joonsoo, B., Hyerim, B., Kang, S., Yeongho, K.: Automatic control of workflow processes using ECA rules. IEEE Transactions on Knowledge and Data Engineering 16(8), 1010–1023 (2004)

11. Russell, N., ter Hofstede, A.H.M., van der Aalst, W.M.P.: Workflow Control-Flow Patterns: Revised Vie (2006), http://www.workflowpatterns.com (last accessed December 14, 2013)
12. Russell, N., ter Hofstede, A.H.M., van der Aalst, W.M.P., Mulyar, N.: Quality metrics for business process models. In: BPM and Workflow Handbook, pp. 179–190 (2007)
13. Fahland, D., Lübke, D., Mendling, J., Reijers, H., Weber, B., Weidlich, M., Zugal, S.: Declarative versus Imperative Process Modeling Languages: The Issue of Understandability. In: Halpin, T., Krogstie, J., Nurcan, S., Proper, E., Schmidt, R., Soffer, P., Ukor, R. (eds.) BPMDS 2009 and EMMSAD 2009. LNBIP, vol. 29, pp. 353–366. Springer, Heidelberg (2009)
14. Michels, G., Joosten, S., van der Woude, J., Joosten, S.: Ampersand, Applying Relation Algebra in Practice. In: de Swart, H. (ed.) RAMICS 2011. LNCS, vol. 6663, pp. 280–293. Springer, Heidelberg (2011)
15. zur Muehlen, M., Indulska, M.: Modeling languages for business processes and business rules: A representational analysis. Information Systems 35, 379–390 (2010)
16. Ross, R.G., Lam, G.S.W.: Building Business Solutions, Business Analysis with Business Rules. Business Rule Solutions, LLC (2011)
17. Spreeuwenberg, S., Healy, K.A.: SBVR's approach to controlled natural language. In: Fuchs, N.E. (ed.) CNL 2009 Workshop. LNCS (LNAI), vol. 5972, pp. 155–169. Springer, Heidelberg (2010)
18. Schwitter, R.: Controlled natural languages for knowledge representation. In: Proceedings of the 23rd International Conference on Computational Linguistics, pp. 1113–1121. Association for Computational Linguistics (2010)
19. Maggi, F.M., Montali, M., Westergaard, M., van der Aalst, W.M.P.: Monitoring business constraints with Linear Temporal Logic: An approach based on colored automata. In: Rinderle-Ma, S., Toumani, F., Wolf, K. (eds.) BPM 2011. LNCS, vol. 6896, pp. 132–147. Springer, Heidelberg (2011)
20. Declare prototype workflow management system, http://www.win.tue.nl/declare/ (last accessed December 14, 2013)
21. Pesic, M., Schonenberg, H., van der Aalst, W.M.P.: Declare: Full support for loosely-structured processes. In: Enterprise Distributed Object Computing Conference 2007, p. 287 (2007)
22. McNeile, A., Simons, N.: Protocol modelling: A modelling approach that supports reusable behaviour abstractions. Software and Systems Modeling 5(1), 91–107 (2006)
23. Wedemeijer, L.: A comparison of two business rules engineering approaches. In: BMSD 2012, pp. 113–121 (2012)
24. McKemmish, S., Acland, G., Ward, N., Reed, B.: Describing records in context in the continuum: The Australian Recordkeeping. Metadata Schema Archivaria 1(48) (2006)
25. van der Aalst, W.M.P., ter Hofstede, A.H.M., Kiepuszewski, B., Barros, A.P.: Workflow Patterns. Distributed and Parallel Databases 14(1), 5–51 (2003)
26. Backhouse, R., van der Woude, J.: Demonic operators and monotype factors. Mathematical Structures in Computer Science 3(4), 417–433 (1993)
27. van Dongen, B., van der Aalst, W.M.P.: A meta model for process mining data. In: CAiSE Conference Proceedings (2005)
28. Ampersand prototype, http://wiki.tarski.nl/ (last accessed December 14, 2013)
29. van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M.: Business Process Management: A Survey. In: van der Aalst, W.M.P., ter Hofstede, A.H.M., Weske, M. (eds.) BPM 2003. LNCS, vol. 2678, pp. 1–12. Springer, Heidelberg (2003)
30. Rozinat, A., Wynn, M.T., van der Aalst, W.M.P., ter Hofstede, A.H.M., Fidge, C.J.: Workflow Simulation for Operational Decision Support. Data Knowledge Engineering 68, 834–850 (2009)

KPIs and Their Properties Defined with the EXTREME Method

Ella Roubtsova[1] and Vaughan Michell[2]

[1] Open University of the Netherlands. 6401DL, Valkenburgerweg 177
Heerlen, The Netherlands
[2] Henley Business School, University of Reading,
Whiteknight, Reading, RG6 6UD, UK

Abstract. Key Performance Indicators (KPIs) are the main instruments of Business Performance Management. KPIs are the measures that are translated to both the strategy and the business process. These measures are often designed for an industry sector with the assumptions about business processes in organizations. However, the assumptions can be too incomplete to guarantee the required properties of KPIs. This raises the need to validate the properties of KPIs prior to their application to performance measurement.

This paper applies the method called EXecutable Requirements Engineering Management and Evolution (EXTREME) for validation of the KPI definitions. EXTREME semantically relates the goal modeling, conceptual modeling and protocol modeling techniques into one methodology. The synchronous composition built into protocol modeling enables traceability of goals in protocol models and constructive definitions of a KPI. The application of the method clarifies the meaning of KPI properties and procedures of their assessment and validation.

1 Introduction

Key Performance Indicators are cumulative measures of system achievements during a given time period. The achievements and the corresponding KPIs are related to system goals. The importance of the correct design of KPIs is clarified by their twofold nature. Namely, KPIs are calculated from the operational data, but they are interpreted at the strategic or tactical levels and often used by authorities to make decisions about the payment for the fulfilled work.

KPIs are usually designed for an industry sector with the assumptions made about the business processes in organizations. However, the business process in individual organizations may deviate from the business process used for the KPI definition. Moreover, the values of KPIs are often derived from the information about several businesses of different sectors. Incomplete assumptions about the business processes used for KPI definitions may result in different interpretation of KPIs. The organizations may become incompatible with respect to KPIs. Incomplete assumptions may leave the room for manipulation of KPI values to achieve better report numbers. The management science indicate this situation as "unreliable" and "plan oriented" KPIs [8].

B. Shishkov (Ed.): BMSD 2013, LNBIP 173, pp. 128–149, 2014.

Therefore, the definitions of KPIs and the completeness of assumptions about the underling processes should be validated. Because of the above mentioned twofold nature of KPIs, validation of their definitions demands both the operational models and the strategic models. The operational models are the executable process models for collecting the cumulative measures during model execution. The strategic models are the goal models for reasoning on KPI properties. The approach for validation of KPI definitions should use the related semantics for the goal models and the process models. The approach should enable building simple and easy changeable executable models. The changeability is needed to correct the assumptions about the business process used for KPI definitions.

In this paper, we propose to use the method called EXecutable Requirements Engineering Management and Evolution (EXTREME) [17] for validation of KPI definitions.

EXTREME is a combination of goal modeling, conceptual modeling and protocol modeling [17]. The synchronous composition built into protocol modeling enables the traceability of goals and concepts in the components of the protocol model and the interpretation of KPIs both in terms of goals and in terms of processes. The execution of the protocol model with the upturn and downgrade business data is used to validate whether the KPIs indicate the upturn and downgrade tendency, and whether the values of KPIs can be manipulated.

We build our work upon the existing methods [16,19,9] presented in Section 2 and use the conceptual basis of other methods. We have found that none of these methods can support the easy changeable executable process models needed for validation of properties of KPIs. We assume that the main reason for that is the semantic incompatibility of the goal modelling approaches and conventional process modelling approaches indicated in [9].

Section 3 describes the EXTREME method that exploits the semantic compatibility of the goal modelling and protocol modelling approaches. The semantics of the protocol modelling approach is described with the emphasis on the model execution that is needed for definition of KPIs.

Section 4 presents the formalization of the definition of KPIs from the protocol model point of view and interpretation of properties of KPIs in terms of this definition.

Section 5 describes the case for validation of KPIs presented in the program "Improving Access to Psychological Therapies (IAPT)" [5]. It reports the results of application the EXTREME method for the case. A triple of the goal, conceptual and executable protocol models is built on the basis of the IAPT document. We discuss the results of validation of properties of KPIs.

Section 6 presents conclusions and future work.

2 Related Work

2.1 Approaches for KPI Modelling

– A Performance Indicator (PI) is formalized in [16] as a concept with a number of attributes: Name, Definition, Type (continuous or discrete), Time

Frame, Scale, Min Value, Max Value, Source (Law, Company policy, Mission Statement), Owner ("the performance of which role or agent does it measure"), Threshold ("the cut-off value separating changes in the value of the performance indicator considered small and changes considered big"), Hardness ("a performance indicator can be soft or hard where soft means not directly measurable, qualitative, e.g. customers satisfaction") [16].
In order to find the values of the attributes, the authors rely on documents, expert knowledge and previous conceptual models. They indicate that it is not easy to find the information about all proposed attributes in the documentation.
The second concept used for the PI formalization is the performance indicator expression. It is "a mathematical statement over a performance indicator evaluated to a numerical, qualitative or Boolean value for a time point, for the organization, unit or agent." [16]. For example, $Response\ Time \leq 48 hours$. The authors suggest to specify the required values of PIs as constraints estimated by experts with respect to a goal. The relations between different PIs are also modelled using the performance indicator expressions.
Let us notice that this formalization does not answer the question how a KPI can be calculated. The authors claim that they need to integrate the performance view with the process, organization and agent-oriented views of the real organizations. However, there is no information about the process semantics used for modelling and no evidence about validation of the PIs using processes. The authors do not involve the process view in the formalization of an indicator. In any case, the authors write about the process views of real organizations, whereas it is often needed to validate the PIs before their implementation in organizations.

- MetricM [19] is another method formalising KPIs. It "is built upon and extends an enterprise modeling approach to benefit from the reuse of modeling concepts and provide relevant organizational context, including business objectives, organizational roles and responsibilities."
The modelling language MetricML used in MetricM "adds essential concepts to modeling performance indicators..." The concept *Indicator* is used to present a KPI. The MetricML *Indicator* metatype is used for modeling its relations to other indicator types, to reference object types representing organizational context and to goal types.
An alternative "attribute" approach, used by MetricM, conceptualizes performance indicator as a (meta-) attribute of metatypes: e.g. "average throughput time" of a business process type or "average number of employees" of an organizational unit type. We partially use these alternative approach for our formalization in the next section. However, MetricM uses declarative models of performance indicators. The models of underlying processes, needed for execution and validation of KPI properties, are not used in MetricM.

The general tendency of two approaches, presented above, is to postpone the validation of the KPI properties to the moment when the process model of the organization is ready.

In this paper, we claim that the early validation of KPI properties on a business process, used for the KPI definition, may eliminate incompleteness of assumptions about the business process and prevent application of unreliable KPIs.

2.2 Goal Modelling Approaches and Modelling of KPIs

All existing approaches agree that the KPIs relate the system goals and processes. However, the process models are not used for validation of KPIs. Let us look what are the reasons for that.

There are goal oriented approaches, such Knowledge Acquisition in autOmated Specification (KAOS) [3], the User Requirements Notation (URN) [6] and i* modelling framework [20]. These approaches relate goals, business concepts and business processes. KAOS applies state machines to model behaviour of concepts. The URN applies a scenario modelling notation called Use Case Maps (UCM) [2]. Both approaches experience problems caused by the semantic incompatibility between the goal models and process models.

Letier at al [9] explain that the synchronous temporal logic used for goal modelling is interpreted over sequences of states observed at a fixed time rate. On the other hand, the conventional process models (UML state machines, use case maps, UML activity diagrams [14], Coloured Petri Nets [7]) use asynchronous temporal logics that are interpreted over sequences of states observed after each occurrence of an event. Thus, the temporal logic operators have very different meanings in synchronous and asynchronous temporal logics.

The process models built in asynchronous approaches accept the recognised messages, events or operation calls even if the state of the model is not appropriate to handle them. In such states, the messages, events or operation calls are kept in queues, bags or buffers to be handled in an appropriate state of the model. As a result, the behaviour model contains many intermediate states that are not justified by goals and declarative requirements. Analysis of intermediate states may be relevant for validation of asynchronous implementation. However, the goals and the KPIs are defined at a different level of abstraction, namely at the tactical and strategic level, i.e. at the level of observable states of the system.

Letier at al. [9] admit that in order to be semantically equivalent to the synchronous KAOS models, the derived event-based behaviour models need to refer explicitly to timing events. In other words, the event-based models should have elements of synchronization.

3 EXTREME: Goal Modelling with Protocol Modelling

Protocol Modelling [12] is an event-based modelling approach with the elements of synchronization needed for relating the goal and process models. The EXTREME [17] method exploits the semantic compatibility of goal and protocol models in order to simplify executable requirements engineering.

The method combines goal modelling, conceptual modelling and protocol modelling into one method to collect all the information needed for reasoning. The goal modelling and conceptual modelling are similar to the KAOS

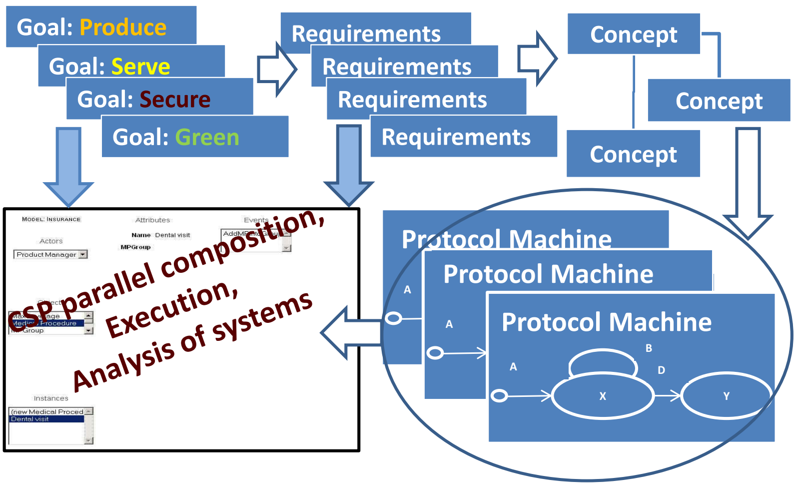


Fig. 1. The EXTREME method

method [3]. These steps are shown in Figure 1 as boxes named Goal, Requirement and Concept. Instead of the UML state machines and activity diagrams used by KAOS, EXTREME models the concept's behaviour as Protocol Machines (Figure 1) composed into a Protocol Models. The Protocol Model is executed using the Modelcope tool [11]. It is shown in Figure 1 as a screen for the user interface. The results of the execution are interpreted in terms of goals and requirements. The interpretation is indicated by the blue arrows.

Goals and concepts are modelled in the declarative way as snapshots of desired system behaviour. Protocol models use a form of synchronous CSP-parallel composition extended with data. They model only the quiescent states of the system that can be easily compared with the states specified for requirements and goals. The goal models and protocol models are semantically coherent. All states of protocol models can be interpreted in the goal semantics. This eases reasoning on models in terms of goals, goal refinement and identification of missing requirements [17].

Although there were many applications of the synchronous CSP parallel composition operator in the architecture description languages [1] and in programming languages [13], only after the extension of this operator for machines with data, made by A.McNeile [12], the operator became practical for business system modelling. The Protocol Modelling proposed in [12] enables coping with complexity of business modelling because the synchronous semantics decreases the data space of models. The process modelling with synchronous semantics solves the semantic mismatch between goal models and process models.

3.1 Protocol Models and Their Execution

Protocol Machine.

A building block of a protocol model is a protocol machine:

$$PM = (E, S_0, S, A, T),\ where$$

- $E = \{e_i\},\ (i = 1, ..., I;\ i, I \in N)$ is an alphabet of event types e_i, i.e. a non-empty finite set of recognized event types coming from the environment. An event type is a tuple of event attributes of different types: $e_i = (A_1^{ei}, ..., A_h^{ei});\ h \in N$. An instance of an event carries data. These data are used to update local storages of protocol machines. An attribute of type *Date* may be used to carry the time moment of event acceptance.
- S_0 is the initial state;
- $S = \{s_j\},\ (j = 1, ..., J;\ j, J \in N)$ is a non-empty finite set of states.
- $A = \{a_k\},\ (k = 1, ..., K;\ k, K \in N)$ is a finite set of attributes of different types. The set can be empty.
- $T = \{t_m\},\ (m = 1, ..., M;\ m, M \in N)$ is a finite set of transitions. $t_m = (s_x, e, s_y),\ s_x, s_y \in S, e \in E$. The set can be empty. The values of attributes are updated only as a result of a transition, i.e. as a result of event acceptance.

Synchronous composition.

In the initial state, a protocol model *PM* is a CSP parallel composition of protocol machines each of which presents a protocol machine type is state *new*. Initially, there are only the machine types serving as patterns for creating instances of protocol machines.

The instances are created by acceptance of events.

At any state, a system model *PM* is a CSP parallel composition of finite set of instances of protocol machines. There are multiple instances of each protocol machine type.

$$PM = \overset{n}{\underset{i=1}{\|}} PM_i = \overset{n}{\underset{i=1}{\|}} (E^{PM_i}, S_0^{PM_i}, S^{PM_i}, A^{PM_i}, T^{PM_i}) = (E, S_0, S, A, T),\ n \in N.$$

A Protocol Model *PM* remains a protocol machine, the set of states of which is the Cartesian product of states of all composed protocol machines [12]:

- $E = \bigcup_{i=1}^{n} E^{PM_i}$ is the set of events;
- $S_0 = \bigcup_{i=1}^{n} S_0^{PM_i}$ is the initial state;
- $S = \prod_{i=1}^{n} S^{PM_i}$ is the set of states;
- $A = \bigcup_{i=1}^{n} A^{PM_i}$ is the set of attributes.

The set of transitions T of the protocol model is defined by the rules of the CSP parallel composition [4]. The rules synchronise transitions T^{PM_i} of protocol machines. Namely, a Protocol Model handles only one event at a time. An event can be accepted only if all protocol machines having this event in their alphabets are in the state where they can accept this event. Otherwise the event is refused.

System Model Execution.

An execution of a protocol model of a system is a sequence of transitions. Processing of an accepted event is instantaneous: it does not take any time. A time moment of event processing may be assigned to the event and saved as a protocol machine attribute. The time moment of the event, that creates a protocol machine presenting a business object, is often useful for calculation of KPIs.

The initial state of an execution may contain any set of instances of protocol machines. At any moment, each protocol machine is situated in one of its states and its attributes have values of their types defined by the history of accepted events. Any state of any execution is quiescent, i.e. it does not change without an acceptance of a new event.

Dependent Protocol Machines. Derived states.

"Two machines having elements of their alphabets in common is not a source of dependency between them" [12]. The dependency means that one protocol machine needs to read the state of another machine to calculate its own state including the attributes. In any quiescent state, a function of this state can be calculated resulting in the extending of the state space of the protocol machine. The state space is extended by the state space of the dependent machines.

Protocol machines can read the state of each other, but cannot change it. This ability of protocol machines to read the state of each other will be used in the next section for the KPI modelling and calculation.

Observational Consistency.

As the CSP composition is applied to all instances of protocol machines in the executable model, it gives to the models the property called observational consistency [10]. This property means that a protocol machine may be added to and deleted from the model or locally changed. The trace behaviour of other protocol machines is not affected by the behaviour of added, deleted or modified protocol machines [10].

4 Definition of KPIs and Their Properties in EXTREME

We use the semantics of a protocol model (section 3.1) to give a constructive definition of a KPI. The definition should specify how a KPI can be derived from a protocol model.

Definition 1. *Let a system be presented as a protocol model PM (section 3.1)*

$$PM = \mathop{\|}_{i=1}^{n} PM_i,\ (n \in N) = \mathop{\|}_{i=1}^{n} (E^{PM_i}, S_0^{PM_i}, S^{PM_i}, A^{PM_i}, T^{PM_i}).$$

A KPI is a cumulative function of the cardinality of a set of selected business objects (presented by protocol machines) and the values of their attributes. The selected business objects give the value **true** *to the selection predicate* $\psi(A^{PM_j}, t, I, V_A)$*. The selection predicate compares the state and attribute values of each protocol machine with the border values of attributes* V_A*, the moment of selection* t *and the time interval* I*.*

$$KPI = f(\left|\bigcup PM_j\right|, \bigcup A^{PM_j}),\ where$$
$$PM_j : (\psi(A^{PM_j}, t, I, V_A) = true);$$
$$(0 \le \left|\bigcup PM_j\right| \le n).$$

An algorithm for calculation of a KPI includes

- *a predicate* $\psi(A^{PM_i}, t, I, V_A)$ *for the selection of a number of business objects (presented as protocol machines) using the time of calculation* t*, time interval* I *and a set of given border values of object attributes* V_A*;*
- *a cycle for selection and counting the number of business objects presenting the state of the model that meet the true value of selection predicate;*
- *a KPI calculation formula which arguments are the number of selected business objects and the cumulative variables depending on the attributes of selected business objects.*

The constructive definition of a KPI clarifies the definitions of its properties. We have chosen a set of properties proposed in [8,15] to formalise them on the basis of our definition and the system protocol model.

1. The first property demands that *a KPI should be in a quantifiable form.* Quantification is an act of selection and counting. Our definition of a KPI shows that selection and counting of instances of business objects (protocol machines) of a particular sort is the way of quantification. The result of selection and counting is used as an argument of the function for calculation of a KPI or for the access to the attribute values of selected objects.
2. The second property says: *A KPI needs to be sensitive to changes.* This property is about the design of the function for a KPI calculation. Our model shows that the argument of the function can fall into two categories: a number of business objects or a variable that depends on values of attributes of the group of selected objects. The changes of the sensed elements in the model should cause the corresponding changes of the KPI. *The sensitivity is the minimum magnitude of the change of the sensed element of the model, required to produce a noticeable value of a KPI.* Constructing a KPIs function as a ratio or a cumulative function may affect the sensitivity.

3. The third desired property states that: *A KPI should be linear.* This simplifies the decision making.
   Our definition shows the possible arguments of the KPI formula. If the sensed argument in the model has been identified, it is the matter of the function choice to make the KPI function linear for this argument.
4. The fourth property is: *A KPI should be semantically reliable.*
   From the modelling perspective, we measure the semantic reliability of a KPI by the number of additional assumptions that need to be made in order to derive the KPI value. If the set of additional assumptions is empty, the KPI is semantically reliable. The next section presents an example of reliable and unreliable KPIs.
5. The fifth property relates the KPIs with goals of the system:
   A KPI should be oriented to improvement, not to conformance to plans.
   For validation of this property on a model, the improvement of the system should be defined from the definition of system goals. The improvement should be related to changes of a KPI in sequential time intervals. In order to test the changes of a KPI, several scenarios need to be executed and populate the model with objects. The model of the system and the system itself should guarantee that the arguments of the KPI formulas are objectively changed by the business process and cannot be manipulated. The executable model can help to identify the scenarios of system execution that lead to the KPI values reflecting both the improvement and the downgrade. The scenarios for manipulation of KPI values can be also identified. A manipulative scenario is a scenario of fraud. Existence of these manipulative scenarios is often caused by the incomplete assumptions about the processes for KPI derivation. It can also indicate the problems with business processes when the roles and the access rights are not specified.
   From the modelling perspective, we measure the improvement orientation of a KPI by the number of manipulative scenarios. If the set of manipulative scenarios is empty, the KPI is oriented to improvement.
   Our practical study illustrates all the properties and shows an example of a plan oriented KPI and an example of an improvement oriented KPI.

5 Case Study

The method for validation of KPIs is shown in Figure 2. The ovals show the input and the output. The boxes depict steps of the method, and the arrows indicate model refinement.

The input for the method application is a document that defines KPIs for a business sector. The KPIs are already designed, and some relevant concepts and steps of the business process are present in the definitions of KPIs. The brief summary of the document [5] is presented below.

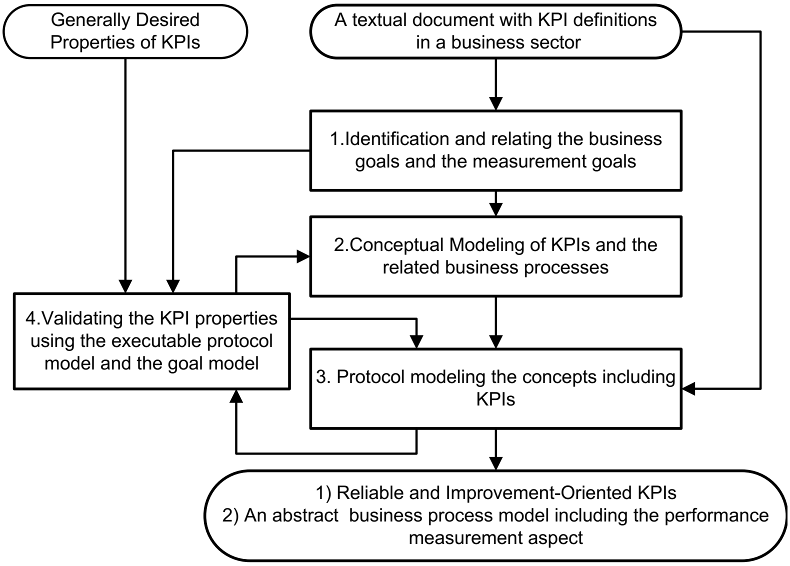


Fig. 2. Using EXTREME for validation of KPIs

KPIs of the Program for Improving Access to Psychological Therapies [5].

- KPI1: Level of Need. It presents the number of people who have depression and/or anxiety disorders in the general adult population. The number presenting population is produced as a result of the Psychiatric Morbidity Survey.
- KPI3a: The number of people who have been referred for psychological therapies during the reporting quarter.
- KPI3b: The number of active referrals who have waited more than 28 days from referral to first treatment/first therapeutic session (at the end of the reporting quarter).
- KPI4: The number of people who have entered psychological treatment, (i.e. had their first therapeutic session) during the reported quarter is related to the concept person.
- HI1: Access Rate. It indicates the rate of people entering treatment from those who need treatment $HI1 = \frac{KPI4}{KPI1}$.
- KPI5: The number of people completed treatment.
- KPI6: The number of people moving to recovery. This number sums up those who completed treatment, who at initial assessment achieve “Caseness” and at the final session - did not.
- KPI6b: The number of people who have completed treatment but were not at “Caseness” at initial assessment.
- HI2: Recovery Rate. It is calculated using the formula $HI2 = \frac{KPI6}{(KPI5-KPI6b)}$.

The IAPT document does not provide information about KPI2 and KPI6a and states that they are no longer collected.

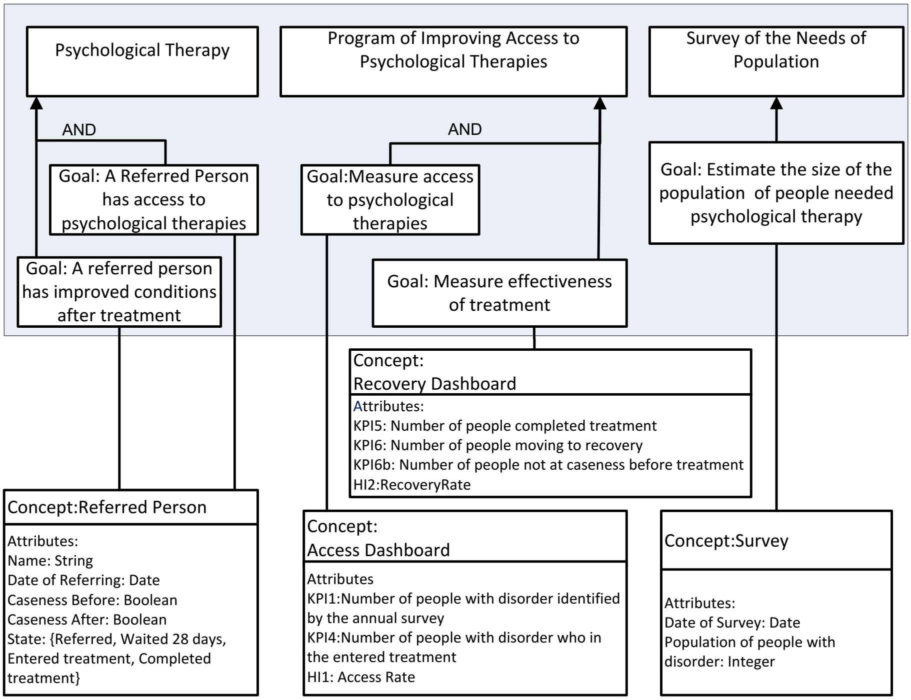


Fig. 3. Goals, Concepts and Protocol Models

The notion of "Caseness" is defined as a result of a condition assessment procedure. The procedure is applied to a referred person. There is no information about the rules of assessment and the values of "Caseness".

Two indicators are called High Indicators (HI). HIs are KPIs calculated from other indicators. In the terminology of Popova and Sharpanskykh [16], the IAPT KPIs can be called PIs and IAPT HIs can be called KPIs. We follow the terminology of the IAPT document [5] and call all indicators KPIs.

5.1 Identification and Relating the Business Goals and the Measurement Goals

In the IAPT document [5], we recognize the goals of measurement:

– *"Measure the access to the psychological therapies."*
– *"Measure the effectiveness of the psychological treatment."*

It is supposed that the underlying business processes:

– *"Estimate the size of population of people needing psychological therapy."*

and guarantee that

– *"A referred person has access to psychological therapy."*
– *"A referred person has improved conditions after treatment."*

The goals indicate three separate business processes: "*Survey of the Needs of Population*," "*Psychological therapy*" and "*Program for Improving Access to Psychological Therapies*".

The upper (grey) part of Figure 3 presents the goal model similar to the models built in Goal-Oriented methods [9]. The boxes are the goals and sub-goals. Goals are refined by the sub-goals that are combined in this case using the logical operator *AND*.

5.2 Conceptual Modelling of KPIs and the Related Processes in the Organization

As in other approaches [16,19], the goals of each process are refined to concepts with attributes. The information about the concepts is taken only from the IAPT document [5]. Concepts are depicted as boxes in the lower (white part) of Figure 3.

The concept *Survey* is the result of the process *Survey of the Needs of Population.*

The concept *Referred Person* is the subject of *Psychological Therapy* mentioned in the goals. We use a generic attribute *State* and identify its possible values of state from the IAPT document. For example, the names of the states of the life cycle of the *Referred Person* are *Referred*, *Waited 28 days*, *Entered treatment* and *Completed treatment.* The results of the condition assessment are modelled by two attributes *CasecessBefore* and *CasenessAfter.* As there is no indication about the type of *Caseness*, we assume that the type is *Boolean.*

In the search of the generic concepts for modeling of KPIs we decided to follow an approach suggested by Strecker et al [19]. We use a concept to present a family of measures for each goal of the measurement. We call such a concept a *Dashboard.* As in the business intelligence, an instance of a *Dashboard* presents a collection of values of measures supporting a particular request.

For example, an instance of the concept *Access Dashboard* shows the current values of indicators *KPI1, KPI3a, KPI3b KPI4* and *HI1*, measuring the access to the therapies. An instance of the concept *Recovery Dashboard* shows the values of the recovery indicators.

The concepts look like UML classes. However, the scarce information from the IAPT document does not allow us to build a complete class diagram and assign roles and relations.

5.3 Protocol Modeling of the Concepts, Including KPIs

In EXTREME, the concepts are modelled as protocol machines (defined in section 3.1).

The Concept *Survey.* is modelled as a protocol machine `Survey`. The protocol model of the `Survey` is described as follows (Figure 4):

```
OBJECT Survey
NAME SurveyName
ATTRIBUTES  SurveyName: String,
            Population:Integer,
            DateOfSurvey:Date
STATES created
TRANSITIONS @new*CreateSurvey=created

EVENT CreateSurvey
ATTRIBUTES  Survey:Survey,
            SurveyName:String,
            Population:Integer,
            DateOfSurvey:Date
```

The metacode, presented above, shows that a protocol machine is a state-transition system. It has its local state described using the keyword `STATES` and `ATTRIBUTES`. A transition from the initial state `@new` is triggered by event `CreateSurvey` which carries data of types `Survey:Survey`, `SurveyName:String`, `Population:Integer`, `DateOfSurvey:Date`.

Each instance of the `Survey` is created by accepting an event `CreateSurvey`. The acceptance of an event `CreateSurvey` brings with its attribute `Population` the number of people who have depression and(or) anxiety disorders and with its attribute `DateOfSurvey` the value of the attribute of the protocol machine `Survey`. Only the `Survey` in state "`created`" can provide the values of its attributes of the `LevelOfNeed` and `Population` for performance indicators.

The Concept *Referred Person*. The set of transitions and the state space of a protocol machine can be split into behaviours for the sake of separation of concerns. For example, the concept *Referred Person* is presented as the protocol machine `Referred Person` that `INCLUDES` behaviours `Treatment` and `Assessment`.

Attributes `CasenessBefore:Boolean` and `CasenessAfter:Boolean` store the results of assessment of the patient's conditions.

```
OBJECT ReferredPerson
    NAME PersonName
INCLUDES Treatment, Assessment
        ATTRIBUTES  PersonName: String, DateOfReferring:Date,
        STATES  referred,  28daysWaited,left
        TRANSITIONS @new*Refer=referred,
                    referred*Leave=left,
                    referred*Wait=28daysWaited,
                    left*Return=referred,
                    28daysWaited*EnterTreatment=28daysWaited
BEHAVIOUR  Treatment
    ATTRIBUTES CasenessBefore:Boolean,
        CasenessAfter:Boolean,
            DateOfCompletion:Date
            STATES   entered, completed,
    TRANSITIONS @new*EnterTreatment=entered,
```

```
                  entered*CompleteTreatment=completed
BEHAVIOUR Assessment
    STATES   assessedBefore, assessedAfter
    TRANSITIONS @new*AssessBefore=assessedBefore,
                assessedBefore*AssessAfter=assessedAfter
EVENT Refer
    ATTRIBUTES ReferredPerson:ReferredPerson, PersonName:String,
            DateOfReferring:Date
EVENT Leave
    ATTRIBUTES  ReferredPerson:ReferredPerson
EVENT Wait
    ATTRIBUTES  ReferredPerson:ReferredPerson
EVENT Return
    ATTRIBUTES  ReferredPerson:ReferredPerson
EVENT EnterTreatment
    ATTRIBUTES ReferredPerson:ReferredPerson,CasenessBefore:Boolean
GENERIC AssessBefore
    MATCHES EnterTreatment
```

Figure 4 shows the protocol machines graphically. Protocol machines look like state machines. However, they have different semantics.

1) The `INCLUDES` relation of protocol machines is shown in Figure 4 as an arrow with a half-dashed end. The `INCLUDES` relation means that for every instance of `Referred Person` the instances of the dependent protocol machines `Treatment` and `Assessment` are created. The behaviours `Treatment` and `Assessment` are equally CSP parallel composed with other protocol machines.

The state space of a `Referred Person` is the Cartesian product of the state spaces of the `Referred Person` and the included behaviours `Treatment` and `Assessment`.

2) Protocol Modelling uses events as elements of interaction between the system and the environment and synchronization of protocol machines. Events are presented as data structures and can carry information. Each transition is labelled with an external event.

An event carries data that are used to update the attributes of an instance of the `Referred Person`. For example, the value of the `DateOfReferring` is entered with event `Refer`. `CasenessBefore` is updated with event `AssessBefore`. `CasenessAfter` is updated with event `AssessAfter`.

3) Protocol machines representing different levels of abstraction are easily composed using event matching mechanism.

For example, event `EnterTreatment` is matched with (considered as) event `AssessBefore` in the behaviour `Assessment`. Event `CompleteTreatment` is considered as `AssessAfter` in the behaviour `Assessment`. This is modeled using the keyword `GENERIC`.[1]

[1] In terminology of Aspect-oriented modelling events can be seen as join points. [10]

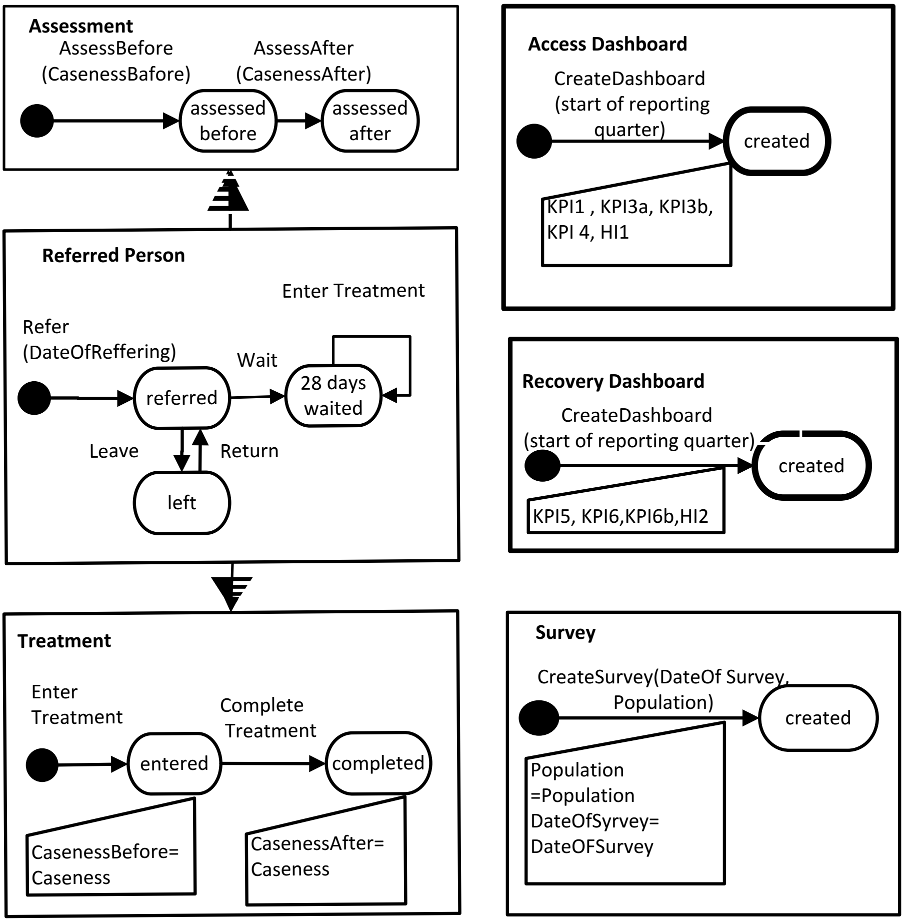


Fig. 4. Protocol Model

4) By submitting events a protocol model is deterministically populated with any number of instances of protocol machines.

5) As a consequence of the CSP parallel composition of protocol machines, the model has only the quiescent states, i.e. the states where the system does not proceed any event. All states can be justified by the system goals. Modelling and reasoning can be focused on the business semantics. Protocol machines in quiescent states can be selected for KPI measurement.

Dashboards and KPIs. The concept *Access Dashboard* and *Recovery Dashboard* are modelled as protocol machines.

```
OBJECT DashboardAccess
NAME  DashboardName
ATTRIBUTES  DashboardName:String,
            StartOfReportingQuarter:Date,
            !LevelOfNeed:Integer,
            !NumberReferredPersons:Integer,
            !NumberReferredPersonsWaited:Integer,
            !NumberOfEnteredTreatment:Integer,
            !AccessRate: Integer,
STATES created
TRANSITIONS
            @new*CreateDashboardAccess=created

OBJECT DashboardRecovery
NAME  DashboardName
ATTRIBUTES  DashboardName:String,
            StartOfReportingQuarter:Date,
            !NumberOfCompletedTreatment:Integer,
            !NumberOfPeopleMovingToRecovery:Integer,
            !NumberOfCasenessPeopleBeforeTreatment:Integer,
            !RecoveryRate:Integer
STATES created
TRANSITIONS
            @new*CreateDashboardRecovery=created

EVENT CreateDashboardAccess
ATTRIBUTES  DashboardName:String,
            DashboardAccess:DashboardAccess,
            StartOfReportingQuarter:Date

EVENT CreateDashboardRecovery
ATTRIBUTES  DashboardName:String,
            DashboardRecovery:DashboardRecovery,
            StartOfReportingQuarter:Date
```

The protocol machines `AccessDashbord` and `RecoveryDashboard` combine two groups of KPIs to monitor the access and recovery. Each dashboard protocol machine reads the state of protocol machines `Survey` and `Referred Person` and derives the values of own attributes presenting KPIs. The derived attributes of dashboard protocol machines, marked by the exclamation symbol "!", represent the corresponding KPIs.

The graphical representation (Figure 4) does not provide all the elements of the model. The complete protocol model of each of concepts *Access Dashboard* and *Recovery Dashboard* include algorithms (or state functions) for calculation of KPIs.

For example, the KPIs of the *Recovery Dashboard* are derived below. Each of the KPIs contains a part selecting objects (*selectInState()*). The selected objects are filtered on the bases of the selection predicate (*if* construction). The filtered objects are counted calculating the value of the corresponding KPI.

```
public class DashboardRecovery extends Behaviour {

public Date getStartOfReportingQuarter(){
    Calendar cal=Calendar.getInstance();
    Date StartQuarter = this. getDate("StartOfReportingQuarter");
    cal.setTime(StartQuarter);
return StartOfReportingQuarter;
}
public Date getEndQuarter(){
    Calendar cal=Calendar.getInstance();
    Date StartQuarter = this. getDate("StartOfReportingQuarter");
    cal.add(Calendar.MONTH, 3);
    Date EndOfReportingQuarter=cal.getTime();;
return EndOfReportingQuarter;
}

// KPI 5 Number of People Completed Treatment in the reported quarter
public int  getNumberOfCompletedTreatment() {
    int NumberOfCompletedTreatment=0;
        Date StartRQ = getStartQuarter();
        Date EndRQ=getEndQuarter();

Instance[] completedTreatment =   selectInState("ReferredPerson", "referred");
    for (int i = 0; i < completedTreatment.length; i++) {
    Date completionDate=completedTreatment[i].getDate("DateOfCompletion");
    String treatmentState = completedTreatment[i].getState("Treatment") ;
    if (completionDate.compareTo(StartRQ)>=0 &&
                completionDate.compareTo(EndRQ)<=0 &&
                treatmentState.equals("completed"))
        NumberOfCompletedTreatment+=1;
        }
    return NumberOfCompletedTreatment;
}

// KPI 6 Number of People Moving To Recovery
public int  getNumberOfPeopleMovingToRecovery() {
    int NumberOfPeopleMovingToRecovery=0;
        Date StartRQ = getStartOfReportingQuarter();
        Date EndRQ=getEndOfReportingQuarter();

Instance[] completedTreatment =   selectInState("ReferredPerson", "referred");
    for (int i = 0; i < completedTreatment.length; i++){
        Date completionDate=completedTreatment[i].getDate("DateOfReferring");
        String treatmentState = completedTreatment[i].getState("Treatment");
        Boolean CA=completedTreatment[i].getBoolean("CasenessAfter");
        Boolean CB=completedTreatment[i].getBoolean("CasenessBefore");
        if (completionDate.compareTo(StartRQ)>=0 &&
                completionDate.compareTo(EndRQ)<=0 &&
                treatmentState.equals("completed") &&
                CB==false && CA==true )
        NumberOfPeopleMovingToRecovery+=1;
        }
return NumberOfPeopleMovingToRecovery;
    }

// KPI 6b NumberOfCasenessPeopleBeforeTreatment from in the reported quarter
public int  getNumberOfCasenessPeopleBeforeTreatment () {
    int NumberOfCasenessPeopleBeforeTreatment=0;
        Date StartRQ = getStartQuarter();
        Date EndRQ=getEndQuarter();

Instance[] completedTreatment = selectInState("ReferredPerson", "referred");
    for (int i = 0; i < completedTreatment.length; i++) {
        Date completionDate=completedTreatment[i].getDate("DateOfReferring");
        String treatmentState = completedTreatment[i].getState("Treatment");
        Boolean CB=completedTreatment[i].getBoolean("CasenessBefore");
        if (completionDate.compareTo(StartRQ)>=0 &&
            completionDate.compareTo(EndRQ)<=0 &&
```

```
            treatmentState.equals("completed") &&
            CB==true)
        NumberOfCasenessPeopleBeforeTreatment+=1;
            }
    return NumberOfCasenessPeopleBeforeTreatment;
        }
```

The strategic indicators combine the KPIs producing rates. For example, the recovery rate combines three indicators:

```
// HI2 Recovery Rate KPI6/(KPI5-KP6b )
public int  getRecoveryRate() {
    int zn=(this.getInteger("NumberOfCompletedTreatment")-
    this.getInteger("NumberOfCasenessPeopleBeforeTreatment"));
    int RecoveryRate=0;
    if (zn==0){ RecoveryRate=0;}
    else {
    RecoveryRate=(100*this.getInteger("NumberOfPeopleMovingToRecovery"))/zn;
    }
return  RecoveryRate;
}
}
```

The complete protocol model can be found in [18]. The Model is executable in the Modelscope tool [11]. The tool generates an interface from the model. The interface is used to submit events, create objects and observe the values of attributes, i.e test the model and validate KPIs.

5.4 Validating the KPI Properties by Using the Executable Protocol Model, Goal and Conceptual Models

Let us analyze if the KPIs in our case study have the desired properties, mentioned in section 4.

1. *The quantifiability of KPIs.*

The KPIs have been modelled as attributes of the concept Dashboard. The protocol model has an algorithm for derivation for each KPI. The algorithm for calculation of a KPI is in the quantifiable form. It contains a predicate for the selection of a number of business objects (presented as protocol machines) using the given time interval and a set of given values of object attributes.

Protocol modelling has predefined select functions. Function

$$selectInState("BehaviourName", "State")$$

returns an array of objects (*"Behaviour name"*) , all of which are in the specified state (*"State"*). Function

$$selectByRef("BehaviourName", "AttributeName")$$

returns an array of objects, all of which have the specified attribute. The select functions enable modelling of quantifiable KPIs.

For example, the *KPI 6b* (the derive function has been shown in the previous section) selects *Referred Persons* initially assessed as Caseness: *CB=true* and then completed treatment within the reporting quarter *RQ*:

$$selectInState("ReferredPerson", "referred")\ \&\ treatmentState.equals("completed")\ \&$$

$$completionDate.compareTo(StartRQ) \geq 0\ \&\ completionDate.compareTo(EndRQ) \leq 0.$$

The number of selected referred persons is counted and gives the value of the indicator. All other KPIs of this case study are similar to *KPI6b.*

2. *The linearity of a KPI* can be tested using the executable model for each KPI. For testing, the model should be populated with objects. The properties of these objects should meet the selection criteria. For example, for the *KPI 6b* the model should be populated with referred persons initially assessed as caseness and completed treatment within the reporting period. The *KPI 6b* should count the amount of such referred persons. The population should also contain the objects that do not meet the selection criteria. Those objects should not be counted by the KPI derivation algorithm.

3. *The sensitivity* should be always validated for strategic indicators presented as rates. The large value of the denominator of a fraction can make the indicator insensitive. The sensitivity of the strategic indicators of our case *Recovery Rate* and *Access Rate* is increased using the percent scale.

4. *We measure the semantic reliability of a KPI* by the number of additional assumptions that need to be made in order to derive the KPI value. If the set of additional assumptions is empty, the KPI is semantically reliable. If the set of additional assumptions is not empty, then the execution and demonstration of the model to the users may be used to validate the assumptions because the users may have more knowledge than the KPI definition document.

For example, in our case, the procedure of assessment of the patient's conditions as *Caseness* is not specified by the IAPT document. We assume that a *Caseness* is a Boolean value coming from the environment. However, in practice, the *Caseness* may, for example, be assessed using a ten-point scale.Therefore, the KPIs, that depend on data assessed via *Caseness*, are not semantically reliable.

The KPI *HI:Recovery Rate* depends on the procedure of testing *Caseness* both before and after treatment:

$$HI2 : RecoveryRate =$$

$$\frac{NumberOfPeopleMovingToRecovery}{(NumberOfPeopleCompletedTreatment - NumberOfCasenessPeopleBeforeTreatment)}.$$

We conclude that this KPI is not semantically reliable.

5. *The improvement orientation of a KPI* is validated for the strategic indicators as they are directly related to the goals of the system.

The meaning of improvement is rarely defined in the KPI documents. It demands extra efforts to guess the notion of improvement by analysing the goals of measurement.

There is a danger of replacing the improvement orientation of KPIs with the plan orientation. In such a case, the "desired" value of KPIs may be achieved through manipulating of numbers of instances in the business process.

Our case study presents examples of both an improvement-oriented KPI and a possibly plan-oriented KPI.

The KPI

$$HI1 : AccessRate = \frac{NumberOfEnteredTreatment}{LevelOfNeed}$$

is an example of an improvement-oriented KPI. It corresponds to the goal: *"A Referred Person has access to psychological therapies."* We assume that the improvement means the positive growth of the ratio of treated people to the people needed treatment.

Modelling shows that the numerator and the denominator of the KPI are the numbers derived from separate processes *Referred Person* and *Survey*. The processes are executed by different organizations, that do not depend upon each other. The *LevelOfNeed* comes from a *Survey*. The *NumberOfEnteredTreatment* is a sum of individually *Referred Persons*. The numbers of objects of separated processes grow independently through the model execution. The manipulation of the numerator and denominator of the KPI is unlikely. Only by the process improvement (for example, by shorten the time of waiting) the higher value of the *Access Rate* can be achieved. Therefore, we conclude that the KPI *HI1:Access Rate* is oriented to improvement.

The KPI

$$HI2 : RecoveryRate = \frac{NumberOfPeopleMovingToRecovery}{(NumberOfPeopleCompletedTreatment - NumberOfCasenessPeopleBeforeTreatment)}.$$

is an example of KPI that may become plan oriented and open to manipulations. For validation of the improvement orientation of this indicator, we use both the goals associated with KPIs and the model of the underlying process. The KPI corresponds to the goal *"A Referred Person after treatment has improved conditions"*.

A person moves to recovery if the assessment of Caseness before treatment is *false (sick)* and after treatment is *true (healthy)*. The improvement corresponds to the growth of the *Recovery Rate* of persons that move to recovery. If the procedures of the *Caseness* assessment and treatment are assigned to the employees of the same organization, who have their interest in high value of *Recovery Rate*, the value of *Recovery Rate* can be manipulated to meet the planned values. This can be done by assessing healthy people as sick before the treatment and sending them for the treatment and/or by assessing sick people as healthy after the treatment. The corresponding scenarios can be shown by model execution.

Validating this property, we conclude that the information in the KPI document is not sufficient to assure the improvement oriented *Recovery Rate*.

One of the possible solutions for improvement of the KPI document may be the following constraint to the business processes of organizations: "The assessment of *Caseness* and treatment should be fulfilled by two independent institutions with different sources of financial support".

Another possible solution is the definition of the rules for *Caseness* assessment, including the assignment of organizations responsible for the treatment and assessment.

The presented examples show that the combination of the goal, conceptual and protocol models, used in EXTREME, provides a useful instrument for validation of KPI properties and leads to discovery of tacit constraints and rules in KPI definitions.

6 Conclusion

In this paper, we have proposed a method for modelling of KPIs and validation of their properties using the approach called EXecutable Requirements Engineering Management and Evolution (EXTREME).

The contribution of this paper is the constructive definition of a KPI. The definition specifies a procedure for calculation of a KPI in models and in information systems. The definition also clarifies the procedures for assessment and validation of properties of KPIs on models.

As a byproduct of the modelling of KPIs with the EXTREME method, an executable model of an abstract organization in an industry sector is produced. In future work, this model can be used for standardization of performance measures in an industry sector and for assessment of applicability of KPIs in real organizations.

References

1. Allen, R., Garlan, D.: Beyond Definition/ Use: Architectural Interconnection. In: Proceedings, Workshop on Interface Definition Languages, Portland, Oregon (1994)
2. Alsumait, A., Seffah, A., Radhakrishnan, T.: Use Case Maps: A Visual Notation for Scenario-Based Requirements. In: 10th International Conference on Human - Computer Interaction (2003), `http://wwwswt.informatik.uni-rostock.de/deutsch/Veranstaltungen/HCI2003/`
3. Dardenne, A., van Lamsweerde, A., Fickas, S.: Goal-directed requirements acquisition. Sci. Comput. Program 20(1-2), 3–50 (1993)
4. Hoare, C.: Communicating Sequential Processes. Prentice-Hall International (1985)
5. Improving Access to Psychological Therapies. IAPT Key Performance Indicator (KPI). Technical Guidance for Adult IAPT Services (2013), `http://www.iapt.nhs.uk/silo/files/iapt-kpi-technical-guidance-201213-v20-.pdf`
6. ITU. Formal description techniques (FDT). User Requirements Notation Recommendation Z.151 (11/08), `http://www.itu.int/rec/T-REC-Z.151-200811-I/en`
7. Jensen, K.: Coloured Petri Nets. Springer (1997)
8. Kueng, P.: Process performance measurement system - a tool to support process-based organizations. Total Quality Management 11(1), 67–85 (2000)

9. Letier, E., Kramer, J., Magee, J., Uchitel, S.: Deriving event-based transition systems from goal-oriented requirements models. Automated Software Engineering Archive 15(2), 175–206 (2008)
10. McNeile, A., Roubtsova, E.: CSP parallel composition of aspect models. In: Proceedings of the 2008 AOSD Workshop on Aspect-Oriented Modeling, AOM 2008, pp. 13–18. ACM, New York (2008)
11. McNeile, A., Simons, N.: (2005), http://www.metamaxim.com/
12. McNeile, A., Simons, N.: Protocol Modelling. A Modelling Approach that Supports Reusable Behavioural Abstractions. Software and System Modeling 5(1), 91–107 (2006)
13. Mount, S., Hammoudeh, M., Wilson, S., Newman, R.: CSP as a domain-specific language embedded in Python and Jython. In: Welch, et al. (eds.) [248], pp. 293–309 (2009)
14. OMG. Unified Modeling Language: Superstructure version 2.1.1 formal/2007-02-03 (2003)
15. Parmenter, D.: Key Performance Indicators, Developing, Implementing and Using Winning KPIs. John Wiley & Sons, New Jersey (2010)
16. Popova, V., Sharpanskykh, A.: Modeling organizational performance indicators. Information Systems 35(4), 505–527 (2010)
17. Roubtsova, E.: EXTREME: EXecuTable Requirements Engineering, Management and Evolution. In: Diaz, V.G. (ed.) Progressions and Innovations in Model-Driven Software Engineering, pp. 65–89. IGI Global (2013)
18. Roubtsova, E.: Protocol Model of the KPIs from the program "Improving Access to Psychological Therapies" (2013), http://www.open.ou.nl/elr/IAPT.zip
19. Strecker, S., Frank, U., Heise, D., Kattenstroth, H.: MetricM: A modelling method in support of the reflective design and use of performance measurement systems. Springer, Information Systems and e-Business Management 10, 241–276 (2012)
20. Yu, E.: Modelling Strategic Relationships for Process Reengineering. Ph.D. Thesis. Dept. of Computer Science, University of Toronto (1995)

Designing Value-Oriented Service Systems by Value Map

Arash Golnam[1], Vijay Viswanathan[2], Christa I. Moser[3],
Paavo Ritala[4], and Alain Wegmann[1]

[1] School of Computer and Communication Sciences,
Ecole Polytechnique Fédérale de Lausanne , Lausanne, Switzerland
{arash.golnam,alain.wegmann}@epfl.ch
[2] Webdoc,
Rue de Genève 17 Lausanne, 1003, Switzerland, Switzerland
vviswanathan@gmail.com
[3] Business School Lausanne,
Route de la Maladière 21, 1022 Chavannes, Switzerland
christa.moser@bsl-lausanne.ch
[4] School of Business, Lappeenranta University of Technology,
Lappeenranta, Finland
paavo.ritala@lut.fi

Abstract. In this paper, we introduce a problem structuring method (PSM) called "Value Map". Value Map is an extension to the Supplier Adopter Relationship Diagram in the Systemic Enterprise Architecture Method (SEAM). Value Map assists in understanding, analysis and design of value creation and capture in service systems. We illustrate the applicability of the Value Map by modeling value creation and capture in the service system of a social networking company called Webdoc. To validate the usefulness of the Value Map, we conducted an empirical study in which we also compared the Value Map to Business Model Canvas, one of the most established methods in business model design. The results of the study show that the Value Map helps business practitioners in understanding and analyzing customer value, customer value creation, and the value capture processes. We conducted an empirical study in which we assessed the usefulness of Value Map and compared it with Business Model Canvas, one of the most established methods in business model design. The results of the study show that the Value Map helps business practitioners to understand and analyze customer value, customer value creation, and the value capture processes.

Keywords: Modeling, Problem Structuring Method (PSM), SEAM, Service Systems, Value Creation and Capture, Value Map.

1 Introduction

In the theories of economic exchange, value was traditionally viewed only from the perspective of monetary transactions between the customers and the organization. Value was perceived to be rooted in goods that were produced by the organization. Once distributed to customers, the value produced was destroyed, or consumed. In the

B. Shishkov (Ed.): BMSD 2013, LNBIP 173, pp. 150–173, 2014.

marketing literature, this perspective is broadly referred to as the Goods-Dominant (G-D) logic, which was prevalent pre-1900s. From the standpoint of the G-D logic, customers played a negligible role in the value creation process. In other words, the organization created what was perceived as valuable to the customer [1] without the involvement of the customer. The underlying objective of the G-D logic is to "maximize operational efficiency and reduce firm costs in order to increase financial profits". Moreover, G-D logic primarily focuses on operand resources (i.e., those resources that are tangible; physical goods) that are manifested in products [2].

In a paradigm shift, the economic exchange model has been augmented and extended to include customers as a fundamental tenet of the value creation process. This shift has led to the emergence of the Service-Dominant (S-D) logic. The G-D and the S-D logic differ in a number of important ways, (see Table 1).

Table 1. G-D Logic vs. S-D Logic

	G-D Logic	**S-D Logic**
Focus	Operand resources; creating goods to be sold	Operant resources; intangible resources (i.e., knowledge and skills)
Goods	The product of value to be exchanged	Seen as intermediaries in service delivery
Service	Intangible output of a good	Service is the foundation of all exchange
Value	Created within organizations	Co-created by organizations and customers

The focus in the S-D logic is on intangibles, competencies, dynamic exchange processes and relationships that are broadly referred to as operant resources. Operant resources have an influence on other resources to create benefit through the service [2].The concept of a good in the G-D logic is the product of value to be exchanged. While, in the S-D logic, a good is merely seen as an intermediary in the delivery of service, broadly viewed as delivery mechanisms for services [3].Furthermore, in the S-D logic, the concept of service is extended beyond a "particular" kind of intangible good (i.e., knowledge and skills) or an intangible output of a good. Instead, service is deemed as the foundation of all exchange (i.e., service exchanged for service) [4]. Finally, the S-D perspective conceptualizes a firm's offerings not as an output, but as an input for the customer's value-creation process. Thereby, instead of viewing value as being created within companies, value is increasingly viewed as being co-created between companies, customers, and other actors within a service system.

Service systems are the arrangement of resources, including people, information, and technology [5]. In service systems, value is perceived as being created in collaboration with the customer. Authors in [6 - 9] argue that in the S-D logic, the supplier is not the sole creator of value, but that value emerges when the customer is involved in the process. Thus, from the S-D standpoint, customers are the eventual locus and the determining party of the value that is created [10]. Authors in [11] suggest that the customer's collaborative role in value creation is what is known as co-creation of value.

Moreover, the S-D logic emphasizes on the subjective and experiential nature of value and thus asserts that value is "uniquely and phenomenologically determined by the beneficiary [3]. Based on this perspective, a distinction is made between value-in-use and value-in-exchange. Value-in-use refers to the specific qualities of the service. These qualities are perceived by users in relation to their needs (i.e., speed or quality of performance, aesthetics, or performance features). Value-in-exchange can be defined as the "monetary amount realized at a certain point in time in exchange" [12]. After value has been created, it is important for the organization to capture this value. Authors in [12] explain that some value may be lost or in some cases, shared with other stakeholders. Value capture, also termed value retention or value appropriation, deals with the amount of exchange value the customer has kept and retained by the organization in the form of profit [13]. From a non-monetary perspective, value capture can be described as the degree to which service quality goals have been met or exceeded [14].

Once value has been (co)-created, the viability of the service system depends on its ability to capture the created value. In other words, the service provider sustains its existence with the value it retains [15]. Thus, it can be asserted that sustainable value (co)-creation and capture is an imperative for viability of service systems. In the service science literature, a number of modeling frameworks provide conceptual tools to support the design of service offerings (see for instance [16-20]). Such modeling frameworks, however, mainly address the design and analysis of value from the customers' perspective and do not sufficiently address service providers' value capture in the service value equation. In general, the same gap can be broadly identified in the service literature, where value (co)-creation has often been emphasized over value capture.

Moreover, there are nonlinearities and feedback structures inherent in the interplay between value creation and capture in service systems. For instance, a slight increase in price, results in the loss of a huge proportion of the market, or, a new service feature can boost the customer base of a service provider. While presenting both conceptual and practical challenges for service providers and service science researchers, this systemic interconnectedness has been glossed over in the service science research.

To tackle the above mentioned research gaps, in this study, we introduce the Value Map; a framework for modeling value in service systems that takes into account both value creation (for and with customers) and value capture (by service providers). The Value Map can be broadly referred to as a problem structuring method (PSM) [21-23] that aims to provide conceptual and practical assistance in analyzing, reconfiguring and designing value in service systems. The modeling constructs and notational elements in the Value Map are derived from a literature review we conducted to gain a new perspective into the structure and the dynamics of value creation and capture.

This paper is organized in the following way. In Section 2, we elaborate on the structure and the results of the literature review we conducted to discover the important concepts relevant to value creation and capture. In order to gain a better understanding, the concepts and their relationships were formalized in 10 algebraic functions and were graphically represented in form of a conceptual model. In Section 3, we introduce the value map and its modeling constructs and notational elements. In Section 4, we present the results of the application of the Value Map to model and improve value creation and capture in a social networking platform called Webdoc. Section 5 includes the related work. In Section 6, we briefly report on the results of an empirical study we conducted to assess the usefulness of the Value Map and to compare it to Business Model Canvas [24] an established method for presenting and designing business models. Finally, in Section 7 we present our conclusions, limitations of research and our future work.

2 The Conceptual Model

In this section, we present the structure and the results of the literature review we conducted on the theoretical frameworks that examine value creation and capture. A literature review can be conducted for a variety of purposes see [25] In this paper, the literature review will help us discover the important concepts relevant to value creation and capture and explore the relationships among these concepts in order to gain a new perspective into the structure and the dynamics of value creation and capture. Thus, the literature review helps us understand the "what" (i.e., the concepts), the "how", (i.e., their relationships) and the "why" (i.e., the rationale behind the selection of the concepts and the perceived relationships among them). According to [26], the "what", "why" and the "how" are the three tenets of a theoretical contribution.

The correct selection of the published materials is a vital element of a literature review. We followed the methodology in [27] and developed a number of criteria for selection of the work to be included in the literature review. The articles we included in the literature review addressed value creation and capture simultaneously, and were indexed by Institute for Scientific Information (ISI). These two criteria led us to a total of around 30 articles. We then derived the key concepts discussed in each article. The concepts were then analyzed and divided into three categories: customer value, customer value creation process, and service provider value capture. Next, for each category, we developed a number of functions that embody algebraic expressions explaining the relationships between the concepts (see Table 2). Having identified the concepts and their relationships, we graphically represented them in form of a conceptual model made up of boxes (i.e., the concepts) and arrows (i.e., their relationships), (see Figure 1). According to [26], "such visual representations often clarify the author's thinking and increase the reader's comprehension". As illustrated in Figure 1, we have marked the three categories of concepts in the conceptual model.

Table 2. The Algebraic Functions capturing relationships between Customer Value, Customer Value Creation Process, and Service Provider Value Capture concepts

Customer Value Conceptualizations	1	Net perceived customer value (NPCV) = (perceived service benefits) – (perceived service costs) *NPCV equals the benefits minus the costs of receiving the service.* [28-30]
	2	Perceived benefits of the service offering = (perceived functional benefits) + (perceived emotional benefits) *The sum of the functional and emotional benefits constitutes the perceived benefits of the service offering.* [28, 31]
	3	Perceived costs of the service offering = (Perceived non-monetary costs) + (Perceived monetary costs) *The costs incurred to the customer who receives the service are divided into two categories: monetary cost and non-monetary costs that can include time, energy, and psychic costs.* [13, 28]
	4	Relative NPCV of the service offering = (NPCV of the service provider's value network offering) – (NPCV of the competing value network's service offering) *Relative net perceived customer value is the net perceived value created by a service provider's offering in relation to the competing offerings.* [13]
Customer Value Creation Process Conceptualizations	5	Service components ⊂ Resources and capabilities (of the service provider and its value network) *Service components are a subset of the resources and capabilities of the service provider and its value network that are manifested in the service.* [13, 32-33]
	6	Service components (of service provider and its value network) ⇒ Service features ⇒ Service value attributes (of service customer) *Service components create some emergent properties for the service, which are noticed by the customer. We refer to these emergent properties of the service as service features. Service features impact the perceived customer value through various value attributes.* [34]
Service Provider Value Capture Conceptualizations	7	NPCV of the service offering ∝ Service providers benefits *The customer's relative perception of value determines the actions the customers undertake, which result in generating more or less benefits for the service provider.* [13]
	8	Net captured value (NCV) of the service provider = (Value captured by the service provider) – (Cost of the service components) *The NCV is the value captured by the service provider minus the costs of the service components.* [13]
	9	(Non-)monetary benefits for the service provider ∝ Value captured by the service provider *The (non)monetary benefits created by the customer for the service provider are proportional to the value captured by the service provider.* [35 -37]
	10	Costs of the service components = (Organizing costs i.e. internal costs of the service provider) + (Opportunity costs i.e. external costs of the suppliers in service provider value network) *The costs of the service components equal the sum of the organizing costs of the service provider, and the external opportunity costs of the suppliers in the value network.* [38,39]

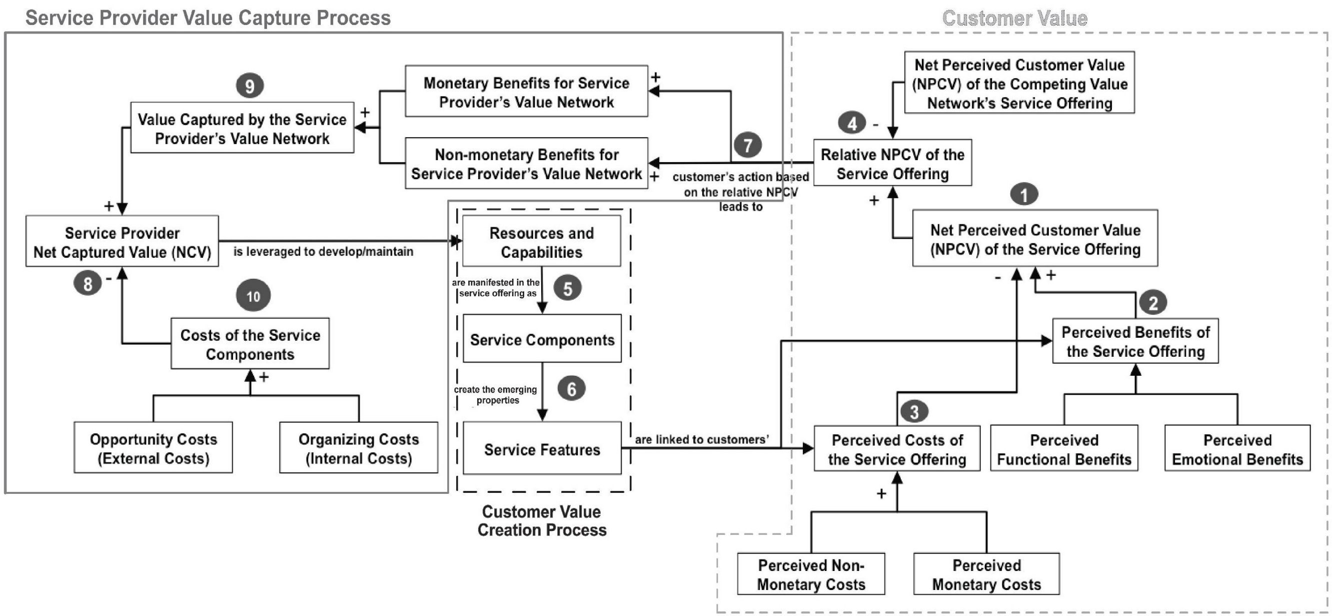


Fig. 1. Graphical representation of value creation and capture concepts

3 The Value Map

Figure 2 represents the actors and their properties in a service system. We refer to this representation as the *Service System Model*. As illustrated, the *Service System* is composed of a *Service Provider Value Network* and *Service Customer A and B*. The *Service Provider Value Network* can be represented as a black-box or a white-box denoted respectively by grey and white colors. In Figure 2, [w], [c] denote whole (black- box) and composite (white-box) representations of the systems and entities. When represented as a black-box we model the *Service,* the *Service Features*, and the *(Non)monetary Benefits* for the *Service Provider Value Network* as its emergent properties. The white-box view of the *Service Provider Value Network* provides insight into the configuration of the value network. Thus, we will be able to view the organizations or the people who compose the value network and their contribution to the service in terms of the *Service Components* they provide. We can also see the *Value Captured* by each of the entities in the value network. As illustrated in Figure 2, the *Service Provider* collaborates with *Organizations A and B* and the *Developer* to create the *Service*. This collaboration is captured in terms of the *Service Components* each of these entities provides. Finally, in the Service System Model we represent the *Service Customers* by modeling the *Attributes* that impact their perception of the service value and the *Actions* the customers take on the basis.

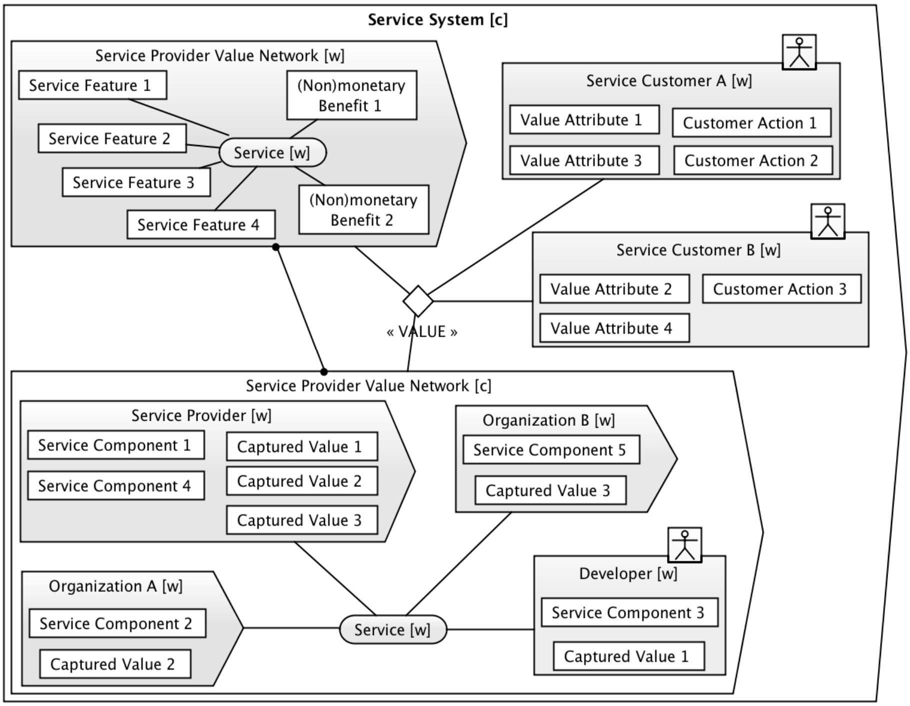


Fig. 2. Service System Model

A generic Value Map is illustrated in Figure 3. As marked in Figure 3, the Value Map embodies customer value, customer value creation and service provider value capture processes (i.e., the three categories of concepts presented in Section 2) in a service system. This is achieved by making the relationships between the actors (i.e., service provider, organizations in the value network, service customer, etc.) and the properties (service components, service features, value attributes, customer actions, etc.) presented in the Service System Model explict. In Table 3, we explain the relationships and their notation in the Value Map.

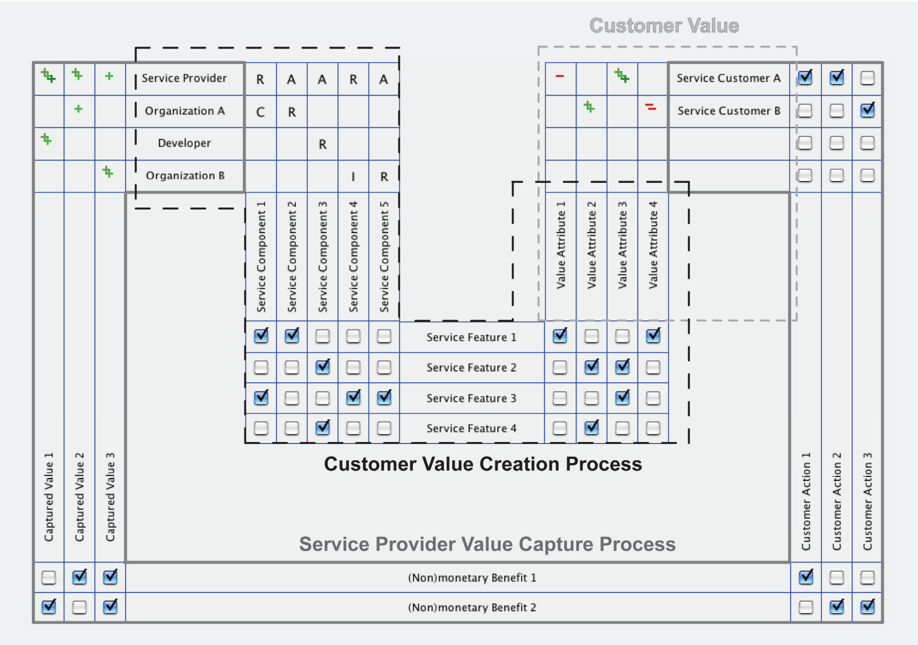


Fig. 3. The Value Map

To map the *Service Provider* and the other entities in the *Service Provider Value Network* to the *Service Components* we use the RACI (Responsible, Accountable, Consulted, Informed) Matrix. As illustrated in Figure 3, the *Service Provider* is responsible for *Service Component 1* and consults *Organization A*. This consultation may reduce the risk of incompatibility between the *Service Components 1 and 2* or ensure the existence of a contingency plan in case an unanticipated scenario arises in the value creation process. The *Service Provider* also contributes to the service through *Service Component 4*. Here, the *Service Provider* makes sure that *Organization B* is kept informed about the progress. The *Service Component 4* provided by *Organization B* may be affected by *Service Component 5* which is provided by the *Service Provider*. Note that these two service componets create the *Service Feature 3*. This sheds light on why *Organization B* needs to be kept up-to-date. In principle, the *Service Provider* is accountable for correct and thorough provisioning of the service components for which other entities are responsible.

Table 3. Relationships and mappings in the Value Map

Relationship	Mapping Notation
- Entities in the value network - Service Components	RACI Matrix
- Service Components - Service Features	☑
- Service Features -Value Attributes	☑
-Value Attributes - Net Perceived Customer Value	+++ Strong Positive --- Strong Negative
- Customer - Customer Actions	☑
- Customer Actions - (Non)monetary Benefits	☑
- (Non)monetary Benefits - Captured Value	☑
- Captured Value - Service Provider's Net Captured Value	☑

As discussed in the previous section, *Service Components* create the *Service Features* that impact the net perceived customer value (NPCV) through the *Value Attributes*. Based on his or her perception of the value of the service offering, the customer takes *Actions*. These *Actions* generate the *(Non) monetary Benefits* for the *Service Provider Value Network*. These benefits are directly linked to the *Value Captured* by each of the entities in the value network. In Figure 3, the *Service Provider* and *Organization A* provide *Service Components 1 and 2* respectively. These two components will create the *Service Feature 1* that negatively impacts the NPCV for *Service Customer A and B* through *Value Attributes 1 and 4*. As shown, the impact is stronger for the *Service Customer B*. Similarly, the *Service Provider* and *Organization B* provide *Service Components 4 and 5* respectively, thereby creating *Service Feature 3*. This service feature has a strong positive impact on the *Service Customer A*'s perception of the service value as captured in Value *Attribute 3*. *Service Customer A* takes *Customer Actions 1 and 2* that contribute to the *(Non) monetary Benefits 1 and 2* thereby realizing and contributing to *Captured Value 1 - 3* for the entities in the *Service Provider Value Network.* As shown, Captured Value 1 has a strong and medium positive impact on the net captured value of the Service Provider and Developer respectively. Other sections of the Value Map can be interpreted the same way.

4 Modeling Value Creation and Capture in Webdoc

In this section, we report on the application of the Value Map as a diagnostic tool to improve value creation and capture in the service system of Webdoc. First we present some information about Webdoc and the motivations underlying the project in which the Value Map was applied. Next, we discuss how customer value attributes were surfaced by means of the data capture and user intelligence tools. Then, we model the creation and capture of value in Webdoc using the Value Map. Finally, we present some strategy implications based on the findings from our modeling process.

4.1 Webdoc

Webdoc is an Internet startup founded in Lausanne in 2009. It currently has offices in Lausanne (headquarters: management, engineering, design, and product), London (business development), Lima (community engagement and support), and San Francisco (business development). Webdoc provides a social network platform on which users can express themselves in a richer, more interactive way than traditional social networks. Specifically, it provides a channel in which existing web content, be it video, audio, images, or text, can be combined with content created using the proprietary rich editor, in a way that requires no technical skills and is easy to share and distribute. These creations are referred to as "webdocs" and can be embedded on any third-party site, including other websites and social networks. Additionally, all webdocs created can be showcased in their relevant category of interest on the Webdoc destination site. The creators have the option to make their webdocs completely private (only users granted explicit permission can view) or public but unlisted (meaning the webdoc will not be featured on the Webdoc site). The service is free to all users with no advertising, currently available in 5 languages (English, French, Spanish, Portuguese, and Russian), and accessible through a variety of platforms including desktop web browsers, mobile device browsers, and native mobile applications.

As the company and user base has grown tremendously in the past 12 months, there has been an increasing need for establishing a better understanding of and improving perceived customer value. The analysis, conception, and subsequent improvement of the value perceived by the customers feed into vital functions of the service and company, including product development, overall strategy, valorization of the company for current and future investment rounds, and optimization of the service. These needs are what triggered the work that has led to the culmination of this project. In the next sections we explain how the value attributes were surfaced and how Value Map improved value creation and capture in Webdoc's service system.

4.2 Surfacing Customer Value Attributes

One of the main challenges in modeling value creation and capture in service systems is surfacing the customer value attributes. This is considered as an important initial step to gain insights into the customers' perceived benefits and costs of adopting the services offered by a service provider. In the context of the project conducted at Webdoc, this step was further sub-divided into two distinct but strongly interconnected fields: data capture and user intelligence.

4.2.1 Data Capture

Broadly speaking, information on customers' perceptions of value and their relative importance can be gathered through direct interaction with customers or customer surveys. Revealed preference methodologies [40] are also used to understand customers' needs and preferences based on their behavior. However, for Internet-based services, the channels through which the service provider can understand its users are very different than those of a traditional service. The overwhelming difference is the radically new interaction paradigm through which service providers and service adopters communicate. For traditional service providers, a wealth of customer data, such as customer demography, is gathered without any explicit effort, simply by the customer's physical presence. On the other hand, for an Internet firm like Webdoc, sophisticated measures need to be put in place to understand even the most fundamental characteristics of its users, such as location, language, gender, and age. Without the application of data capture tools it would almost be impossible to answer basic questions such as "Who are the service customers?" "How frequently do they use the service?" "How do service customers access the service?" "How much do they use the service for?" To answer such questions, a number of service providers offer web analytics packages. These are third party, off-the-shelf solutions that can be customized to varying degrees, and are provided for a cost ranging from free to tens of thousands of dollars a month. There also exists the possibility for every Internet company to custom-build its own web analytics and data capture solution. In the context of this project, the latter was the first solution considered, but was quickly discarded due to its infeasibility.

4.2.2 User Intelligence

Data capture contributes to the decision processes in Internet-based services by providing macro-level information. User intelligence tools, however, provide a much more nuanced perspective at the micro level, which sacrifices on breadth of data for depth. The fundamental motivation of the application of user intelligence was the need for product development insight. While numeric metrics such as overall visitors, logged in users, views of a particular page, and so on are certainly invaluable, they are more useful in measuring the effectiveness (or ineffectiveness) of a feature post-change than they are in suggesting what changes might be needed in the first place. Thus, user intelligence provides data that is more prescriptive. This data is complementary to the descriptive data derived from the data capture tools.

User intelligence applications offer various analytical and intelligence tools such as *heat maps* and *user recordings*. Heat maps are screenshots of the website showing the spatial distribution of clicks over the screen space that offer important product insight, as they show what links and content garner the highest level of attention from the audience. User recordings are an attempt to recreate individual user sessions by aggregating mouse movement, keyboard activity, scrolling and navigation, and clicks into a video.

Some advanced user intelligence applications provide the possibility of creating a test environment in which a random sample of participants execute tasks that are predefined based on the demographic and technical requirements. Upon completion of the tasks, a questionnaire is automatically generated, which is filled out by the participant. The key aspect is that while performing the tasks, the entire user screen is recorded, along with an audio stream for the live commentary of the participants. The application of data capture and user intelligence tools provided invaluable assistance in surfacing the customer value attributes.

4.3 Modeling Value Creation and Capture in Webdoc

In this section, we apply the Value Map to represent value creation and capture in Webdoc's service system. To this end, first we shed some light on how Webdoc can capture value as a service provider. Next, we analyse the value for Webdoc's customers. Finally, we show how the Value Map resulted in improving value creation and capture in Webdoc.

4.3.1 Value Capture by Webdoc

Internet-based companies, in particular, social networking platforms such as Webdoc follow a free business model [24] this means these companies do not charge the customers for service they offer. Thus, to sustain their existence these service providers rely solely on the non-monetary benefits from their customers. These non-monetary benefits in the case of Webdoc include: number of users, volume of activity per user, and time spent on the platform per user. Such non-monetary benefits can result in value capture for Webdoc by:

- *Increasing the valuation of the company in case of an initial public offering (IPO) or acquisition.* As an example, Instagram, the online photo sharing service provider, was acquired by Facebook in April 2012 for $300 million in cash and 23 million shares of common stock. The deal was worth $1 billion at the time. Before the acquisition, Instagram announced that more than 5 billion photos had been shared through its mobile apps [41].

- *Securing funding by venture capitalists (VCs).* Most start-ups rely on funds from external sources such as VCs [42]. The non-monetary benefits listed above are among the determining factors for VCs to make a decision to invest or to continue investing in a start-up company like Webdoc.

- *Monetization through advertisement.* Another possibility for Internet-based services is generating revenues by authorizing the presence of advertisements on their webpages. Advertisement-based monetization is one of the main revenue streams for internet-based service providers. The number of visitors, their activity volume and the time they spend on a website are the main criteria for businesses or individuals to choose a website on which they place their advertisements.

4.3.2 Value for Webdoc's Customers

To improve the NPCV, first an understanding of different customer categories of Webdoc needs to be established. Two main categories of customers are identified: first-time and return visitors. When a first-time visitor uses Webdoc's services again, he becomes a return visitor.

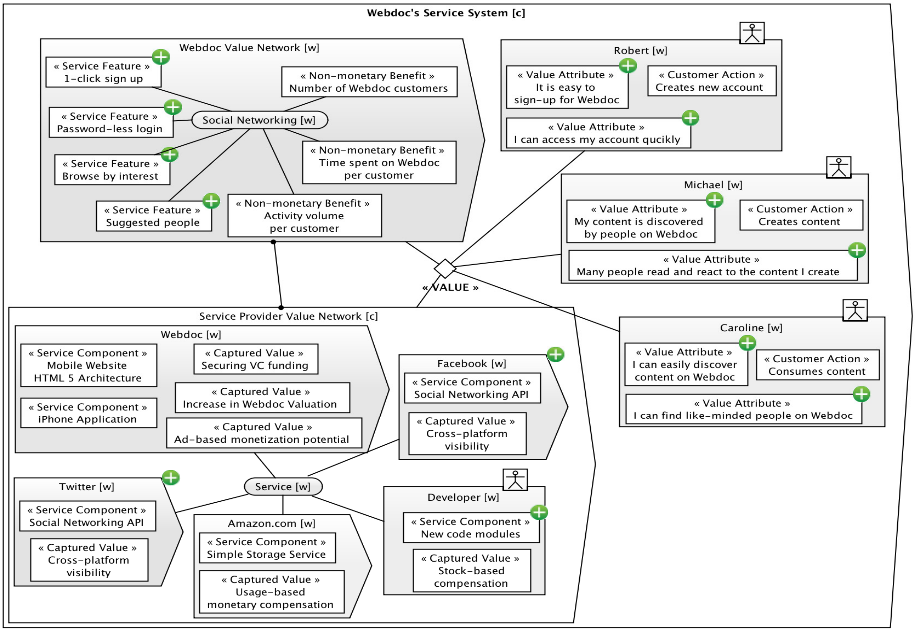


Fig. 4. Webdoc's service system, new actors and properties

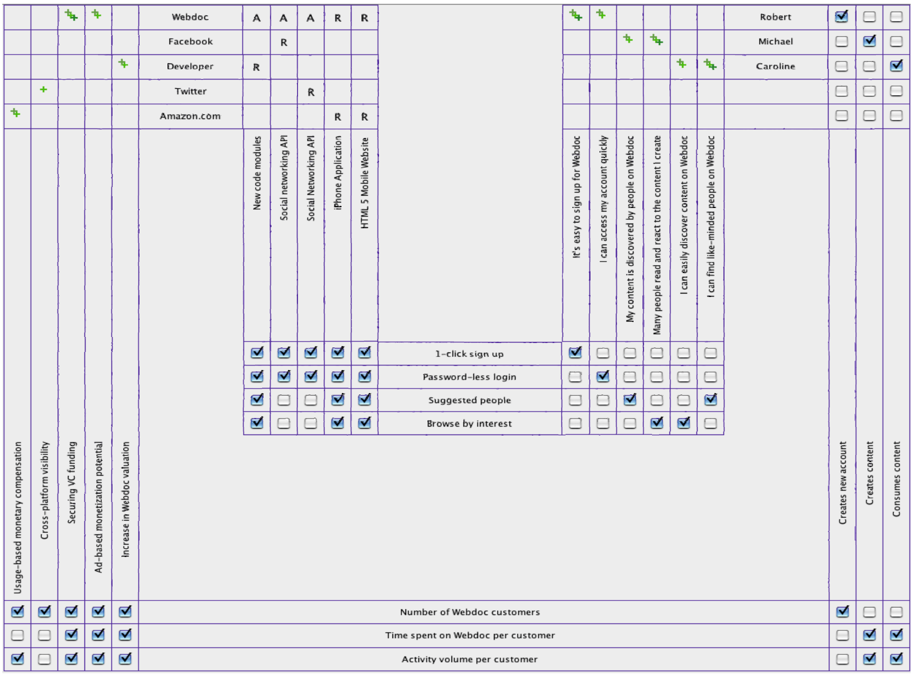


Fig. 5. The Value Map: capturing value creation and capture in Webdoc's service system

The return visitors are divided into two main categories: content creators and content consumers.

- *Content creators.* This category of customers creates or curates the content on Webdoc. Curation is the process of sorting content created by others on the web and presenting it in a meaningful and organized way.

- *Content consumers.* These customers consume and react to the contents created by others. The reaction takes place in form of *sharing, liking,* or *commenting on* the contents.

It should be stated that these customer categories are not mutually exclusive.

As discussed in Section 2, the NPCV is impacted by the benefits and the costs of the service. As Webdoc offers its service to the customers free of charge, it would have been intuitive to focus on the service benefits to improve the NPCV. However, the findings from the data capture and user intelligence step revealed a number of improvement opportunities concerning the non-monetary costs of the service. As outlined in Table 2, these non-monetary costs include but are not limited to time, energy, and psychic aspects of adopting a service. Time cost is the aggregate of the durations the service customer needs to invest in order to be able to use the service. Energy cost is the sum of the effort that needs to be spent. Psychic cost is the most abstract of all – the cognitive stress undergone by the customer in using the service.

Data gathering and user intelligence aided us in mapping these three categories of non-monetary costs onto their corresponding value attributes (i.e., non-monetary costs of the service):

- Filling out the sign up form and discovering content and people increased the time and energy costs of the Webdoc's customers.

- Remembering passwords incurred Webdoc customers with high psychic costs.

4.3.3 Improving Value Creation and Capture in Webdoc

Figure 4 shows Webdoc's service system. The new actors and their properties are marked with the plus sign. In the Value Map in Figure 5, we illustrate how these novelties result in improving value creation and capture in Webdoc service system. We explain these changes in the following sections.

4.3.3.1 Reducing Clicks to Signup. Initially, an unregistered visitor needed to search for the sign up button, click it, and then fill out a form to complete the process. To save the visitor's time, the application programming interfaces (APIs) from social networking websites, Facebook and Twitter were integrated in the home page of Webdoc as shown in Figure 6. This way, the first-time visitors could sign up with one click without filling out the sign up form. The return visitors could also use their Facebook or Twitter credentials to connect to Webdoc.

As illustrated in the Value Map in Figure 5, Twitter and Facebook provide Webdoc with the APIs as the service components. These APIs along with the *New code modules* provided by the *Developer* result in the service features *1-click sign up* and *Password-less login*. These two new features create the following two value attributes

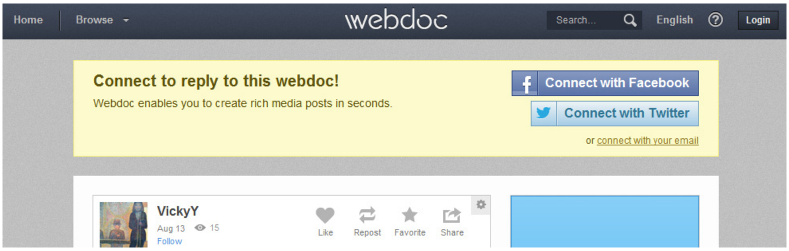


Fig. 6. Reducing clicks to sign up

for Robert who is a first-time visitor: *It's easy to sign up for Webdoc* and *I can access my account quickly*. These features along with the rest of the benefits of Webdoc convince Robert to *create a new account* (i.e., customer action) thereby increasing *number of Webdoc customers* increases. The rise in the number of customers contributes to *Increase in Webdoc valuation, Ad-based monetization potential* and *Securing VC funding*. As stated in Section 4.3.1, these are the main ways Webdoc can capture value. We can also see that Ad-based monetization potential is not as important as the other two value attributes. The number of Webdoc customers also gives Twitter and Facebook *Cross-platform visibility,* which can contribute to their web presence.

The two new service features improved the NPCV by reducing the time and energy costs associated with filling out the sign up form as well as the psychic costs of the remembering passwords. Introduction of these features increased the number of new accounts created on Webdoc. Moreover, nearly two months after their implementation, over 80% of the users were logging in to Webdoc using their Twitter and Facebook accounts.

4.3.3.2 Welcome Workflow. To facilitate discovering content and content creators, a welcome workflow was designed, see Figure 7.

As shown in the Value Map, the welcome workflow, captured in the *New code modules* service component, resulted in the creation of two new service features: *Browse by interest* and *Suggested people*.

Caroline is a content consumer. The two features help her in *finding like-minded people* and *discovering content on Webdoc*. Michael, a customer who creates content on Webdoc, benefits from these two features as *his content is discovered by people on Webdoc* and *many people read and react to the content he creates*. These value attributes form a self-reinforcing positive feedback loop. Michael creates content, which is discovered by Caroline. Caroline consumes Michael's content and reacts to it by commenting or liking or reposting his content. This motivates Michael to create or curate even more content. This virtuous cycle increases *time spent on Webdoc per customer* and *activity volume per customer*. These two non-monetary benefits contribute to Webdoc's value capture the same way as the *number of Webdoc customers*. When Webdoc's valuation increases some value is also captured by the developer who receives stock-based compensation. Finally, operating on the basis of a pay-per use pricing mechanism, Amazon.com also captures some value when the number of the customers and the activity per customer increase.

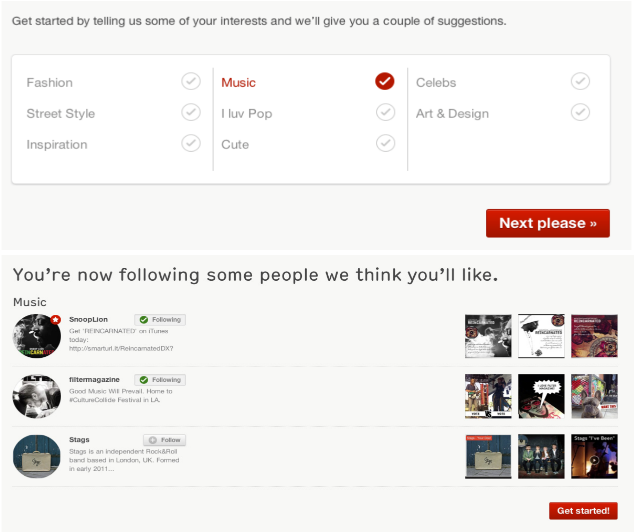


Fig. 7. Welcome workflow

The new features created by the welcome workflow reduce the energy and time costs pertinent to discovering people and content. The introduction of these two features contributed to the 250% increase in the log in rate of the Super Users, those customers of Webdoc who visit the website at least three times a week.

4.4 Strategy Implications for Webdoc

Over the past few months, Webdoc has improved the value its customers perceive from the services it offers, by reducing the non-monetary costs associated with its services. These improvements have resulted in an increase in the number of customers, the activity and the time spent by each customers. However, similar to any growth pattern there are limits to this growth.

We suggest that Webdoc should also become a platform to promote the work of the artists who are not famous. A young Sci-Fi writer, an unknown musician or a semi-professional painter can be the potential new actors in Webdoc's service system. These people should not be merely seen as customers. In fact, they should be taken into account as parts of the Webdoc's value network. Expanding the value network results in the creation of a bigger pie for all the organizations and people involved and results in creating more value for customers.

5 Related Work

The Value Map is an extension to the SAR (Supplier Adopter Relationship) Diagram in [43, 44]. The Service System Model is based on the System Diagram [45]. The

SAR and the System Diagram are parts of the Systemic Enterprise Architecture Methodology (SEAM) [46]. SEAM was designed from the ground up with general systems principles and serves to analyze and to assist in the design of business and engineering strategies. Developed at Ecole Polytechnique Fédérale de Lausanne (EPFL), SEAM has been used for teaching [44] and consulting [47].

In developing the Value Map, we are also inspired by the House of Quality [48], a quality improvement method, derived from Quality Function Deployment (QFD). We integrate the Strategy Canvas [49] as a part of the Value Map. Strategy Canvas is a diagnostic framework for strategy development. It enables an organization to visualize the competitive factors and the current state of play of those factors within a market place and to compare the organization's offering with those of the industry in general.

Business Model Canvas [24] is a strategic management tool, that assists in the development of new, and improvement of existing business models. It is widely recognized as one of the most established methods for business model design and innovation. The canvas represents value creation and capture in business models by nine building blocks: key partners, key activities, key resources, value propositions, customer relationships, channels, and customer segments. Business Model Canvas is one of the most established methods in the academia and industry for business model design, development and improvement.

6 Empirical Study

In this section, we elaborate on the empirical study that we conducted to assess the usefulness of The Value Map. In this study, we also compared the Value Map with Business Model Canvas [24]. Business Model Canvas is a strategic management tool, that assists in the development of new, and improvement of existing business models. The canvas represents value creation and capture in business models by nine building blocks: key partners, key activities, key resources, value propositions, customer relationships, channels, and customer segments. Business Model Canvas is one of the most established methods in the academia and industry for business model design, development and improvement.

We organized three workshops attended by 14 participants from Iran. The participants belonged to various industry sectors, such as automotive parts manufacturing, power generation, pharmaceutical and investment. They all held executive and senior management positions in their companies and had a minimum of 8 years of experience.

The first workshop lasted for 6 hours. In the first part of this workshop, we discussed business modeling and problem structuring in organizational decision processes. We also explained the theoretical concepts such as value creation and capture in business models. Then, we familiarized the participants with business model canvas and its nine building blocks. Next, we provided an example of a business model represented by the business model canvas. It should be noted that 5 participants were already familiar with the Business Model Canvas and/or applied it in representing a business model.

In the second workshop that also lasted for 6 hours, we presented the Value Map, its underlying theoretical perspectives along with the example provided in this chapter. The participants were then divided in four groups. Each group decided on a business idea. In the groups where the members were from the same industry background a real business idea was chosen. The groups represented their business ideas first with the business model canvas and then the Value Map. We acted as facilitators during the sessions and answered to the participants' questions. When the models were completed and presented by the groups. During the presentation of the models, we provided feedback on the models to each group.

The third workshop was held a week after the second workshop lasted for three hours. We had asked the participants to reflect upon the usefulness, practicality and the potential merits of the Value Map and its positioning with respect to the Business Model Canvas. In the workshop, which lasted nearly 3 hours, we debriefed the participants. Some of them had tried to apply the Value Map in their organizations and shared their experiences with us.

At the end of the second workshop, a survey questionnaire was distributed among the participants. As shown in Figure 8, the participants had to specify whether they strongly disagree, disagree, are undecided about, agree or strongly agree with the nine statements. The first statement was on the importance of value creation and capture in the business model of an organization. Statements 2-5 were derived from the proposed future work by Osterwalder in [24: 141] "I propose that future work on business models includes testing the following hypotheses developed on the basis of the interviews with business practitioners:

• A business model ontology based visualization tool can help business practitioners more quickly understand a business model and the relationships behind its elements.
• A business model ontology based tool creates a common langue to address business model issues and in this regard improves communication between business practitioners.
• Discussing business model issues with a business model ontology based tool (to understand business models) has an impact on discussion quality."

In statements 6-9, we compare the Value Map and the business model canvas.

The results show that around 90% of respondents either agree or strongly agree with the statements. There is no proposition that is strongly rejected.

As presented in Table 4, all participants either agree or strongly agree on the importance of value creation and capture in the business model of an organization. This sheds light on the practical relevance of the topic of the research for the industry practitioners who participated in the survey.

Based on the responses to Statements 2 and 3, almost all of the participants found the Value Map a tool that can help them in understanding and analysis of value creation and capture by creating a common language that enables them to jointly represent, discuss the as-is situation and envision the to-be situation of the customer value creation and capture processes in the organization's business model. Over 85% of the participants agreed that the common language created by the Value Map

SURVEY OF VALUE MAP, A PROBLEM STRUCTURING METHOD FOR VISUALIZING THE BUSINESS MODEL OF AN ORGANIZATION

Name and Surname		Respondent's Signature
Organization		
Type of Business		
Organizational Position		
Years of Experience		

1	Strongly disagree	2	Disagree	3	Undecided	4	Agree	5	Strongly Agree

		1	2	3	4	5
1	Value creation and capture are central to an organization's business model.					
2	The Value Map helps business practitioners to understand and analyze customer value; customer value creation, and the value capture processes in an organization's business model.					
3	The Value Map creates a common language for business practitioners that enables them to jointly represent, discuss the as-is situation and envision the to-be situation of the customer value creation and capture processes in the organization's business model.					
4	The Value Map facilitates and improves the communication among the business practitioners within the organization. This improvement in communication positively impacts the quality of the discussions regarding the business model of the organization.					
5	The Value Map is a useful visualization tool that contributes to managerial decision making processes of business practitioners about the value creation and capture in an organization's business model.					
6	Business Model Canvas helps identify the building blocks of an organization's business model.					
7	The two additional building blocks in the Value Map (i.e., the competing offers and, the service/product features) are useful and/or necessary for representing an organization's business model.					
8	The Value Map models the inter-relations, inter-connections, linkages and the dynamics between the building blocks of a business model.					
9	The Value Map complements and augments the Business Model Canvas by aiding the business practitioners in representing the necessary building blocks of business model of an organization and their inter-relations and interconnectedness.					

Questionnaire distributed among the participants in the business modeling workshop in the Ministry of Industry, Mine and Trade Dec-28-2012 – Tehran, Iran

Page 1 of 1

Fig. 8. The questionnaire

Table 4. Results of the survey - frequency of responses for statements in percentage

	Strongly Disagree	Disagree	Undecided	Agree	Strongly Agree
Statement 1	0.0%	0.0%	0.0%	28.6%	71.4%
Statement 2	0.0%	0.0%	0.0%	85.7%	14.3%
Statement 3	0.0%	7.1%	0.0%	57.1%	35.7%
Statement 4	0.0%	14.3%	0.0%	71.4%	14.3%
Statement 5	0.0%	14.3%	7.1%	64.3%	14.3%
Statement 6	0.0%	7.1%	0.0%	57.1%	35.8%
Statement 7	0.0%	0.0%	0.0%	28.6%	71.4%
Statement 8	0.0%	7.1%	0.0%	50.0%	42.9%
Statement 9	0.0%	0.0%	0.0%	50.0%	50.0%

improves the communication among the business practitioners within the organization and thereby positively impacts the quality of the discussions by surfacing the practitioners' implicit assumptions regarding the business model of the organization. Finally, the majority of the participants came to the conclusion that the Value Map is a useful visualization tool that contributes to managerial decision making processes of business practitioners about the value creation and capture in an organization's business model.

Based on the results reported in Table 4, the participants found the Business Model Canvas a useful method for identifying the building blocks of a business model. As stated earlier Business Model Canvas represents the business model by nine building blocks: key partners, key activities, key resources, value propositions, customer relationships, channels, and customer segments.

In the Value Map, we represent two building blocks additional to the ones conceptualized by the Business Model Canvas, namely: competing offerings and product/service offer features. The participants either strongly agree or agree that these two additional building blocks are useful and/or necessary for representing an organization's business model. The participants also concluded that Value Map models the interconnections between the building blocks of a business model, whereas the Business Model Canvas just aims at identifying these building blocks. Finally based on the responses to Statement 9 we can state that participants thought that the Value Map can complement and augment the Business Model Canvas by representing the necessary building blocks of business model of an organization and their inter-relations.

Table 5 illustrates a summary of the statistics of the survey. As shown, the average (mean) of the responses to the statements is 4.27 which means the participants either agree or strongly agreed with all the statements. The standard deviation for all the statements is 0.67, which is relatively small with negative skewness values.

Table 5. Summary of the statistics of the survey

	Mean	Median	STD	Skewness
Statement 1	4.71	5.00	0.469	-1.067
Statement 2	4.14	4.00	0.363	2.295
Statement 3	4.21	4.00	0.802	-1.482
Statement 4	3.86	4.00	0.864	-1.361
Statement 5	3.79	4.00	0.893	-1.035
Statement 6	4.21	4.00	0.802	-1.482
Statement 7	4.71	5.00	0.469	-1.067
Statement 8	4.29	4.00	0.825	-1.583
Statement 9	4.50	4.50	0.519	0

To sum up, based on the debriefing in the third workshop, the participants found the Value Map a useful visualization tool that can contribute to the decision processes that require competitor analysis, understanding customer needs and preferences and the features of the product or service that needs to be designed or improved to meet and fulfil the customer needs. Some of the participants stated that the Value Map can be of great value for cross functional teams and when applied for organizational diagnosis. The discussions with the participants also revealed a number of improvement opportunities in terms of adding a quantitative model, simplifying the graphical representation, and the parsimoniousness of the conceptualizations.

Some of the improvement opportunities mentioned by the participants are already taken into account in the instantiation of the Value Map in the www.tradeyourmind.com online platform. For instance, the inclusion of the quantitative models that can generate numerical analyses of various value creation and capture strategies is part of the platform. The step-by-step model generation wizard embedded in the www.tradeyourmind.com platform also facilitates the development and the presentation of the Value Map. We will try to address the remaining points in our future work.

The participants also commented on the relationship between the Value Map and the Business Model Canvas. They were unanimous that the representations created by Business Model Canvas can be used as an input to the Value Map. In other words, the Value Map makes explicit the relationships between the building blocks of a business model represented in the Business Model Canvas.

7 Conclusions

In this paper, we introduced the Value Map as a problem structuring method (PSM) that aids in conceptualization and representation of value creation and capture in service systems. The Value Map is grounded in the theoretical insights from economics, management science and (services) marketing literature, drawing principally upon work from the past two decades on value creation and capture, including theories, frameworks, constructs, and other models. We illustrated the usability and applicability of our framework by modeling value creation and capture in Webdoc's service system. We also briefly presented the results of a survey conducted to assess the usefulness of Value Map and compare it with Business Model Canvas.

This research suffers from a number of limitations. We used data synthesized in a single case study to illustrate the applicability of the Value Map. Despite the fact that the data for the case study was gathered from a project we conducted in a company, we believe we need to apply the Value Map in several other contexts to be able to fully assess the practical relevance of its representations. Thus, in our future work we will focus on applying the Value Map in a number prospective business cases. This will definitely result in a better evaluation of the applicability of the Value Map.

The second limitation of this research concerns the empirical study we conducted to evaluate the usefulness of the Value Map. The fact that all the participants in the survey were from Iran and the relatively small sample size limit the generalizability of the findings of our research. To tackle this limitation, the same study should be conducted among executives and managers from different countries.

Lastly, the articles based on which the conceptualizations underlying the Value Map were developed are not exhaustive. Despite the fact that we synthesized over 30 well-cited articles on value creation and capture that were to the best of our knowledge seminal to the field, some relevant work still may not have been included in the review of the literature. Inclusion of such articles can bring in new modeling constructs or fine-tune and improve the existing constructs in the Value Map. Refining our conceptualizations based on the existing work that has not been included in the study will also be a part of our future work.

References

1. Prahalad, C., Ramaswamy, V.: Co-creation Experiences: The Next Practice in Value Creation. Journal of Interactive Marketing 18(3) (2004)
2. Vargo, S.L., Lusch, R.F., Akaka, M.A.: Advancing Service Science with Service-Dominant Logic - Clarifications and Conceptual Development. In: Handbook of Service Science, pp. 133–156 (2010)
3. Vargo, S.L., Lusch, R.F.: Service-Dominant Logic: Continuing the Evolution. Journal of the Academy of Marketing Science 36(1), 1–10 (2008)
4. Vargo, S.L., Lusch, R.F.: Evolving to a New Dominant Logic of Marketing. Journal of Marketing 68(1), 1–17 (2004)
5. Vargo, S.L., Maglio, P.P., Akaka, M.A.: On Value and Value Co-creation: A Service Systems and Service Logic Perspective. European Management Journal 26, 145–152 (2009)
6. Sphorer, J., Maglio, P.: The Emergence of Service Science: Toward systematic service innovations to accelerate co-creation of value. Production and Operations Management (2008)
7. Grönroos, C.: Service marketing: a study of the marketing function in service firms (In Swedish with an English summary). Diss. Helsinki and Stockhold, Hanken Swedish School of Economics, Marketing Technique Centre and Akademilitteratur, Helsinki (1979)
8. Grönroos, C.: Adopting a service logic for marketing. Marketing Theory 6(3), 317–333 (2006)
9. Grönroos, C.: Service logic revisited: who creates value? And who co-creates? European Business Review 20(4), 198–314 (2008)
10. Ballantyne, D., Varey, R.J.: Creating value-in-use through marketing interaction: the exchange logic of relating, communicating and knowing. Marketing Theory 6(3), 335–348 (2006)
11. Gummesson, E.: Exit services marketing - enter service marketing. Journal of Customer Behaviour 6(2), 113–141 (2007)
12. Sandström, S.E., Kristensson, P.: Value in use through service experience. Managing Service Quality 18(2), 112–126 (2008)
13. Lusch, R.F., Vargo, S.L.: The Service-Dominant Logic of Marketing: Dialog, Debate, and Directions. M. E Sharpe, Armonk (2006)
14. Lepak, D.P., Smith, K.G., Taylor, M.S.: Value Creation and Value Capture: A Multilevel Perspective. Academy of Management Review 32(1), 180–194 (2007)
15. Bowman, C., Ambrosini, V.: Value Creation Versus Value Capture: Towards a Coherent Definition of Value in Strategy. British Journal of Management 11(1), 1–15 (2000)
16. Parasuraman, A., Zeithaml, V.A., Berry, L.: A Conceptual Model of Service Quality and Its Implications for Future Research. Journal of Marketing 49, 41–50 (1985)
17. Ritala, P., Hurmelinna-Laukkanen, P.: What's in it for me? Creating and appropriating value in innovation-related coopetition. Technovation 29(12), 819–828 (2009)
18. Gordijn, J., Akkermans, J.: Value-based requirements engineering: Exploring innovative e-commerce ideas. Requirements Engineering 8(2), 114–134 (2003)
19. Weigand, H., Johannesson, P., Andersson, B., Bergholtz, M.: Value-based service modeling and design: Toward a unified view of services (2009)
20. Pijpers, V., Gordijn, J.: e 3 forces: Understanding Strategies of Networked e 3 value Constellations by Analyzing Environmental Forces Advanced Information Systems Engineering, 188-202 (2007)

21. Yu, E.S.K.: Towards modelling and reasoning support for early-phase requirements engineering. Paper Presented at the The Third IEEE International Symposium on Requirements Engineering (1997)
22. Weigand, H.: Value EncountersñModeling and Analyzing Co-creation of Value. Software Services for e-Business and e-Society, 51–64 (2009)
23. Mingers, J., Rosenhead, J.: Problem structuring methods in action. European Journal of Operational Research 152(3), 530–554 (2004)
24. Rosenhead, J.: What's the problem? An introduction to problem structuring methods. Interfaces 26, 117–131 (1999)
25. Rosenhead, J., Mingers, J.: Rational analysis for a problematic world revisited. John Wiley and Sons, Chichester (2001)
26. Osterwalder, A., Pigneur, Y.: Business model generation: A handbook for visionaries, game changers, and challengers. Wiley, New York (2010)
27. Hart, C.: Doing a literature review: Releasing the social science research imagination. Sage Publications Limited (1999)
28. Whetten, D.A.: What constitutes a theoretical contribution? Academy of Management Review 14(4), 490–495 (1989)
29. Baker, M.J.: Writing a literature review. The Marketing Review 1(2), 219–247 (2000)
30. Kotler, P.: Marketing management, millennium ed. (2000)
31. Day, G.S.: Market – Driven strategy: Processes for Creating Value. The Free Press, New York (1990)
32. Huber, F., Herrmann, A., Morgan, R.E.: Gaining competitive advantage through customer value oriented management. Journal of Consumer Marketing 18(1), 41–53 (2001)
33. Grönroos, C.: Service Management and Marketing. European Journal of Marketing 15, 3–31 (2000)
34. Kothandaraman, P., Wilson, D.T.: The Future of Competition: Value-Creating Networks. Industrial Marketing Management 30(4), 379–389 (2001)
35. Moller, K., Svahn, S.: Role of Knowledge in Value Creation in Business Nets*. Journal of Management Studies 43(5), 985–1007 (2006)
36. Pynnonen, M., Ritala, P., Hallikas, J.: The new meaning of customer value: A systemic perspective. Journal of Business Strategy 32(1), 51–57 (2011)
37. Nelson, R.R., Winter, S.G.: An evolutionary theory of economic change. Belknap Press (1982)
38. Allee, V.: Value network analysis and value conversion of tangible and intangible assets. Journal of Intellectual Capital 9(1), 5–24 (2008)
39. Ulaga, W.: Capturing value creation in business relationships: A customer perspective. Industrial Marketing Management 32(8), 677–693 (2003)
40. Masten, S.E., Meehan, J.W., Snyder, E.A.: The costs of organization. Journal of Law, Economics, and Organization 7(1), 1 (1991)
41. Blomqvist, K., Kylaheiko, K., Virolainen, V.M.: Filling a gap in traditional transaction cost economics: Towards transaction benefits-based analysis. International Journal of Production Economics 79(1), 1–14 (2002)
42. Carson, R.T., Flores, N.E., Martin, K.M., Wright, J.L.: Contingent valuation and revealed preference methodologies: Comparing the estimates for quasi-public goods. Land Economics, 80–99 (1996)
43. Indvik, L.: Facebook Completes $730 Million Instagram Acquisition (2012), http://mashable.com/2012/09/06/facebook-instagram-acquisition-complete/ (retrieved May 2013)

44. Bhide, A.: The origin and evolution of new businesses. Oxford University Press on Demand (2000)
45. Golnam, A., Regev, G., Ramboz, J., Laprade, P., Wegmann, A.: Systemic Service Design: Aligning Value and Implementation. In: Morin, J.-H., Ralyté, J., Snene, M. (eds.) IESS 2010. LNBIP, vol. 53, pp. 150–164. Springer, Heidelberg (2010)
46. Wegmann, A., Julia, P., Regev, G., Perroud, O., Rychkova, I.: Early Requirements and Business-IT Alignment with SEAM for Business. Paper presented at the 15th IEEE International Requirements Engineering Conference (RE 2007), Delhi, India (2007)
47. Rychkova, I., Regev, G., Le, L.-S., Wegmann, A.: From Business To IT with SEAM: The J2EE Pet Store Example. Paper presented at the 11th IEEE International Conference on Enterprise Distributed Object Computing, EDOC 2007 (2007)
48. Wegmann, A.: On the systemic enterprise architecture methodology (SEAM). Paper presented at the International Conference on Enterprise Information Systems, Angers, France (2003)
49. Wegmann, A., Regev, G., Loison, B.: Business and IT alignment with SEAM. Proceedings of the 1st International Workshop on Requirements Engineering for Business Need, and IT Alignment, Paris, France (2005)
50. Clausing, D., Hauser, J.R.: The house of quality. Harvard Business Review 66, 63–73 (1988)
51. Kim, W.C., Mauborgne, R.E.: Blue Ocean Strategy: How to Create Uncontested Market Space and Make Competition Irrelevant. Harvard Business Press (2005)

Author Index